Relation-ontology drive topic classification

Hao, Qi

Awarding institution:
King's College London

Relation-Ontology Driven Topic Classification

Qi Hao

Department of Informatics
King's College London

This thesis is submitted for the degree of
Doctor of Philosophy in Computer Science

King's College London April 2020

I would like to dedicate this thesis to my loving parents, and my husband.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Qi Hao
April 2020

Acknowledgements

When I look back the past four years, I would say that doing a PhD is full of challenges and uncertainties. Find the way to handle it was a long journey. I am sincerely grateful to many people for all their support to help me pass through this journey. I would like to express my greatest gratitude and appreciation to my supervisors, Dr. Jeroen Keppens and Dr. Odinaldo Rodrigues for their valuable guidance, constructive feedback and comments from the very beginning till the completion of this PhD. Without their passionate participation and input, continual faith and encouragement, this work could not have been completed so easily.

My profound gratitude to my parents for providing with continuous encouragement and unfailing support through my years of study and through the process of research and write up of this thesis. My special thanks to Dr. Chih-Han Chen, my loving husband, for providing his valuable support and comments through the process of this work.

Currently, the entire world is struggling against the virulent pandemic COVID-19. Each of us is affected, either overtly or covertly. I would like to give my support and gratitude to the NHS at the most critical time in its history. We are all in this together. I believe we will overcome all the difficulties and embrace a brand new world.

Abstract

Conventional topic models have been extensively used to extract topics in documents for topic classification. For example, Latent Dirichlet Allocation (LDA) is a topic modelling technique that produces a probabilistic model based on word co-occurrence, for the purpose of text classification. However, it is very challenging for these topic models to accurately capture the semantical information of the topics because they focus on the occurrences of words in text rather their meanings and context. They ignore the fact that words may have multiple meanings and that different words may have the same meanings. In addition, they ignore the semantical structures in text, such as the relationship between words in their context. This PhD thesis proposes a novel topic classification technique that uses ontological information and the relationship between words to provide a more accurate topic model for topic classification. Firstly, LDA is extended to use the semantic concepts from an ontology to help capture some of the possible semantical meanings of the words appearing in the documents. The topic model allows topics to be defined more generally in terms of ontological concepts rather than words and this captures the semantical meaning of the words more accurately. In order to capture the relationship between words in the context, this work also introduces a new entity-based algorithm for multiple-relation extraction from unstructured text. The new algorithm uses standard Natural Language Processing (NLP) techniques to analyse unstructured text. The algorithm offers clear performance advantages over conventional single-relation extraction techniques and verb-based techniques. Finally, the extracted structured relations were incorporated with the ontology-driven topic model, resulting in what we called a relation-ontology driven topic classification technique. This topic model allows the topics to be defined more accurately in terms of relations between ontological concepts rather than word co-occurrence. This captures the semantical meaning and semantical structures in text. Our classification approach can be combined with a self-training procedure to reduce the amount of manually classified data required. The classification performance of these topic models was compared against several variations of existing techniques on four widely used datasets. The results show that the inclusion of the ontology component and the contextual relationships help to reduce the training time by nearly quarter whilst achieving the highest accuracy overall in the classification.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Computational linguistic research has leveraged recent increases in computational power to collect and analyse data written in natural language with unprecedented breadth, depth and scale [Lazer et al., 2009]. In daily life, countless new articles and texts are written and published in news outlets, scientific journals, conference proceedings and online websites all across the world. In addition, electronic communication between people through e-mail, text messaging and social media is now an important and common aspect of modern life. According to IBM, around 80% of all information is unstructured, with the text being one of the most common types of unstructured data [Schneider, 2016]. Unstructured text is an effective means of disseminating information to humans. However, to fully comprehend all the information embedded in this form within larger communities or across collections of related disciplines can be very difficult or even impossible. Therefore, researchers started to look beyond traditional approaches for textual data processing in order to recognise and extract hidden patterns and relationships implicitly contained in neighbouring fields of research areas.

An important analysis task in the computational linguistic field is modelling and classify documents into categories based on different topics. By doing so, humans can focus on texts that are most relevant to their current concerns. The fact that manual topic extraction is time-consuming and not easy to scale makes automating this process important. Automatic *topic classification* problems have been widely studied and addressed in many areas over the last few decades. With recent breakthroughs in Natural Language Processing (NLP) and text mining, many researchers are now developing applications that leverage text classification methods. Automatic topic classification provides several advantages:

- Scalability: Manually analysing and organising documents is time-consuming and scales poorly. Automatic topic modelling and classification can easily analyse millions of documents at a fraction of cost. By automate text classification, humans can

structure information such as e-mails, legal documents, medical reports, web pages, chat conversations, and social media messages in a fast and cost-effective way.

- Timeliness: Some critical situations, such as public relation crises on social media, require identifying documents as soon as possible in order to facilitate a prompt response. Automatic topic classification can make accurate analyses in real-time so that critical information can be identified instantly, and actions can be taken right away.

- Consistency: It is very likely for humans to make mistakes during manual annotation due to distractions, fatigue and boredom, which can generate inconsistent criteria. In contrast, automatic topic classification applies the same criteria to all of the documents, thus reducing errors with centralised topic classification.

Topic classification can be applied to a wide range of tasks. For example, it can empower product features and humans inadvertently interact with on a daily basis (such as e-mail spam filtering [Dada et al., 2019]). It can be used to analyse a broad range of documents such as short texts (e.g. tweets, headlines) or much larger documents (e.g. customer reviews, media reports, legal contracts). Topic classification research has been applied successfully in various domains, such as sentiment analysis on social media [Li et al., 2016], medical coding [Zhang et al., 2018a], legal documents [Lu et al., 2011], human behaviour modelling [Hsu and Chiu, 2017], as well as personalised recommendation systems [Wang and Wong, 2013].

Medical coding is an area of healthcare applications where topic classification can be highly valuable. It aims to assign medical diagnoses to specific class values obtained from a large set of categories. In different stages of real-life diagnosis and treatments, such information needs to be available instantly throughout the patient-physician encounters [Lauría and March, 2011]. Topic classification can also be used to analyse Medical Subject Headings (MsSH) and Gene Ontology (GO) [Trieschnigg et al., 2009]. However, these texts are presented in an unstructured or narrative form with ambiguous terms and typographical errors, which makes it difficult to annotate manually and real-time. Automatic topic classification can automatic summarise and extract structured and interpretative knowledge from all kinds of unstructured medial reports for each patient.

Analysing *legal documents* is also an essential application for topic classification. Huge volumes of legal information and documents have been generated by government institutions. The categorisation of these documents is the main challenge for the lawyer community. Automatically extracting this information and classifying them based on topics can help not only lawyers but also their clients [Turtle, 1995]. For example, five basic categories to classify the law are: Constitutional law, statutory law, treaties, administrative regulations, and the common law [Bergman and Berman, 2016].

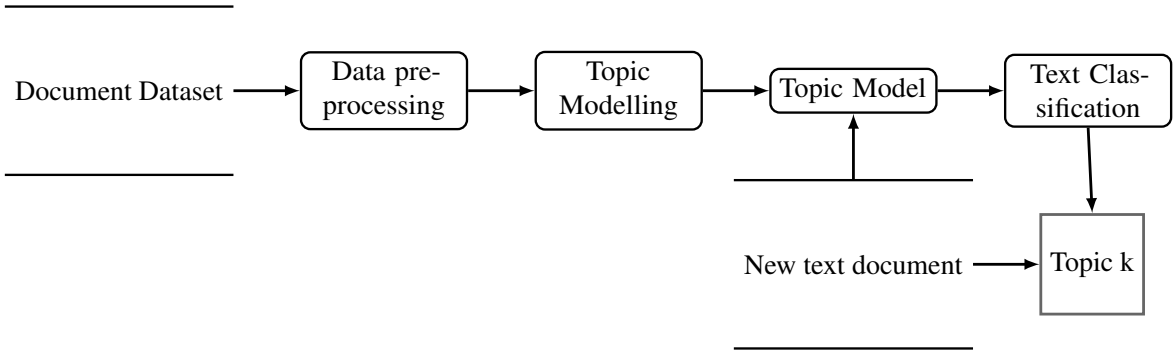


Fig. 1.1 Typical procedure of topic classification

Topic classification problem normally can be solved by two steps: 1)*Topic modelling*: employ a statistical model to summarise text documents into abstract topics; 2) *Text classification*: using the obtained topic model to train a classifier to identify topic terms and classify texts. Figure 1.1 shows a typical procedure of *topic classification*. Given a document dataset, some standard NLP techniques are performed for data pre-processing. Then they are fed into a *topic modelling* procedure to obtain a statistical topic model, which is a mixture of topics. Such a topic model can be used to train a classifier to perform *text classification*. With the topic model, a new text document can be summarised into abstract topics and then be classified into a specific topic by the trained classifier.

Naturally, a reliable topic model would result in high accuracy of text classification, regardless of the classifiers. Thus, topic modelling provides an effective framework for extracting the symbolic representation from unstructured text data. Latent Dirichlet Allocation (LDA) [Blei et al., 2003] is one of the most commonly used *topic modelling* techniques [Li et al., 2016, Hsu and Chiu, 2017, Burkhardt and Kramer, 2018]. LDA is a probabilistic model that infers probability distributions from frequency statistic. However, the symbolic representation of topics which treating words as self-contained tokens can only reflect the co-occurrence relationships of words. The fact that it can not accurately consider the hidden semantical information, such as semantical meanings of words and semantical structures of texts, results in limited performances of the topic models they produce [Campbell et al., 2015b]. Without considering the semantical meanings of words in the text, conventional LDA ignores the fact that words may have multiple meanings and that different words may have the same meaning. However, only consider semantical meanings of words may introduce too many irrelevant information, results in the topic model becoming too general. In order to only consider the relevant concepts in the context, the semantical structures in texts need to be considered. Recently, distributed word representations with neural network language model (NNLM) [Bengio et al., 2003] have significantly improved the performances for

NLP and Machine Learning (ML) tasks [Otter et al., 2020, Khan et al., 2020]. Traditional word embedding frameworks, such as word2vec [Mikolov et al., 2013b] and the GloVe toolbox [Pennington et al., 2014], represent the target word based on a slide window of adjacent words, which is insufficient to capture the entire contextual semantical information, especially when dealing with a small corpus. In addition, the quality of word embedding has a significant influence on topic modelling. Most word embedding methods get word embedding from external corpora, which is inaccurate for word expression, and words that are not included in external word embedding are ignored [Fu et al., 2016]. Recently, new word embedding language models relying on neural network architectures and machine learning frameworks have significantly improved the performance of various NLP tasks, such as BERT [Devlin et al., 2018], GPT and GPT-2/3 [Radford et al., 2018, 2019, Brown et al., 2020], and XLMs [Lample and Conneau, 2019] (they are described in more detail in Section 2.1). They are pre-trained unsupervised on large datasets with over a billion parameters. They can be easily applied in different NLP tasks across domains. By utilising these pre-trained language models, existing state-of-the-art topic modelling approaches are able to include contextual semantics and achieve good classification results [Liu et al., 2019, Peinelt et al., 2020].

In addition, LDA models produced by supervised techniques outperform those produced by unsupervised techniques [Li et al., 2016, Hsu and Chiu, 2017, Burkhardt and Kramer, 2018]. However, supervised techniques require a large amount of manually classified training dataset, which can be very costly to produce [Ko and Seo, 2009]. And as a result, these training datasets are usually small, while larger training datasets not only assure better generalisation but also provide better accuracy.

The main objective of this PhD thesis was to develop a topic model for automated text classification that can address these shortcomings of existing work. The desired topic model can interpret topics more accurately in terms of their semantical meanings and semantical structures. The desired topic classification can be trained with less manually classified data in a faster process. To reach this aim, the following specific objectives and questions are posed.

- **Research question (RQ1):** *Could a topic model considering the semantical meanings of the words achieve better results?*

  We developed an ontology-driven topic classification method with LDA topic model. With a new dataset consisting of raw text documents, we first project a document into a topic matrix by including ontological concepts. Each topic is represented by ontological concepts of words instead of words themselves. By incorporating the ontological concepts, we aim to construct a topic model that can consider the

semantical meanings of texts rather than solely depends on their symbolical meanings. This topic model with ontology allows us to illustrate topics more specifically with knowledge bases so that it can perform the modelling independently of the particular set of words.

- **Research question (RQ2):** *Can a method be developed to capture semantical structure in unstructured texts?*
  We developed an entity-based relation extraction algorithm to extract multiple relations from single sentences. This algorithm aims to extract structured relationships embedded in sentences and documents so that unstructured documents are projected into relation matrices. These extracted relations enable us to capture the semantical structure in texts.

- **Research question (RQ3):** *Could a topic model considering both the semantical meanings and semantical structures of texts achieve better results?*
  We developed a relation-ontology driven topic classification method with LDA topic model. By combining the ontology-driven topic model and the extracted relations, we finally project documents into a relation-ontology topic matrix. Here, each topic is represented by relations between ontological concepts. This topic model is improved by leveraging not only semantical meanings of texts but also semantical structures embedded in the texts.

- **Research question (RQ4):** *Can some techniques be employed to accelerate the training process of a topic classifier and reduce the required amount of manually classified training data?*
  We employed a self-training process to reduce the required amount of manually classified data. Such a self-training process can enlarge a small amount of pre-classified training data and achieve a semi-supervised learning process. In addition, we performed the computing process into a distributed cloud computing service to further accelerate the training process of the topic classification.

Figure 1.2 shows a top-level follow of our research work. Given a document dataset, the proposed topic model with ontology is able to summarise them into a statistical topic model. Instead of relying on external word embedding toolkits to include the semantical meanings of words, we introduced an intermediate concept variable into LDA, resulting in a knowledge-based approach. This ontology-driven topic model is sufficient to capture the semantical meanings of words, regardless of the size of the corpus. By employing the proposed entity-based algorithm, structured relation information can be extracted from

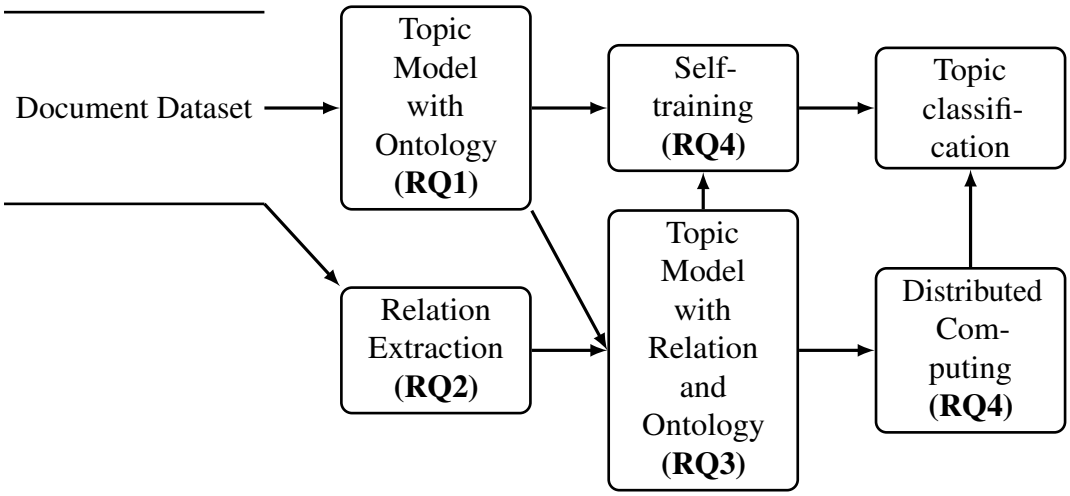


Fig. 1.2 Top-level follow of the PhD research

the document dataset. By combining the ontology knowledge and the extracted structured relations, the obtained relation-ontology topic model is able to consider both the semantical meaning and semantical structures in texts. In addition, a self-training algorithm is introduced and combined with both ontology-driven topic model and relation-ontology driven topic model, aiming to reduce the required amount of manually classified data. A distributed cloud computing platform is also employed to further accelerate the training process of the construction process of the topic model. Finally, the proposed topic model with relation and ontology combined with the self-training algorithm and distributed computing platform is used to train a classifier to perform text classification.

The proposed relation driven topic classification and the entity-based relation extraction algorithm can be evaluated separately using various datasets. Therefore, we performed two experiments for each methodology and described them their corresponding chapters. In order to evaluate the performance of the relation-ontology driven classification, which combines both the relation driven topic classification and the entity-based relation extraction algorithm, a third experiment was performed. For evaluation purpose, all experiment of the proposed work will be presented in this thesis as follows:

- Experimental Setup: we describe the experimental setup. We explain the partition of training sets and testing sets, some commonly used toolkits and the computing environment.

- Datasets Used in the Analysis: we describe the datasets used in evaluation experiments, including the reason for choosing such dataset and basic information about the dataset (number of instances, example of instance, partition of training sets and testing sets).

- Experimental Results: we present the experimental results and discuss the performances of different approaches.

Here, we describe the main contributions of this PhD research:

- Incorporate ontology knowledge with LDA for topic classification to consider semantical meanings of unstructured texts, resulting in a knowledge-based ontology-driven topic classification method. We evaluated the ontology-driven topic classification method using four commonly used datasets against several variations of state-of-the-art methods, such as Term Frequency - Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Even though our knowledge-based OLDA approach achieves lower accuracy of classification compared against word embedding with LDA [Fu et al., 2016](78.01% against 80.94% on 20 Newsgroups dataset) and [Liu et al., 2019] (78.01% against 83.05% on 20 Newsgroups dataset), our approach is superior to theirs in two ways: 1) it does not rely on external word embedding toolkits; 2) it is able to deal with small corpus. Comparing against state-of-the-art knowledge-based approach [Allahyari and Kochut, 2015], the inclusion of the ontology component and logistic regression increase the accuracy of classification employed by between 3 and 7 percentual points (depending on different datasets). It also reduces the construction time of the topic model by nearly half.

- Extract structured relations from unstructured texts using a rule-based algorithm, we call entity-based relation extraction algorithm, to capture the semantical structures of unstructured texts. In order to evaluate the proposed algorithm, two datasets from different domains were used, one from the biomedical domain and the other from the general domain. Comparing against conventional single-relation extraction algorithms, our proposed extensions achieved at least 17 percentual points increase in precision and at least 27 percentual points increase in recall. Our entity-based algorithm outperforms existing state-of-the-art work in datasets from different domains: for the biomedical domain, the F-score is increased by 0.15 [Khordad and Mercer, 2017]; for the general domain, the F-score is creased by 0.04 [Wang et al., 2016]. Our algorithm also outperforms the state-of-the-art algorithm based on Transformers [Eberts and Ulges, 2019]: the F-score is increased from 0.7147 to 0.955. As a side contribution of this work, we also created a new dataset from PubMed for relation extraction task in the biomedical domain that we call PubMed600.

- Combine the ontology knowledge and the extracted structured relations with LDA for topic classification to consider both the semantical meaning and semantical structures

in texts. The result knowledge-based relation-ontology driven topic classification was evaluated using four datasets against the previous ontology-driven topic classification method. Comparing against word embedding with LDA [Fu et al., 2016], our knowledge-based relation-ontology driven approach increases the accuracy of classification by 0.61% (81.55% against 80.94% on 20 Newsgroups dataset). Comparing against state-of-the-art knowledge-based approach [Allahyari and Kochut, 2015], the inclusion of the relationship component increases the accuracy of classification by between 5 and 13 percentual points (depending on different datasets). In addition, the inclusion of the self-training process and the distributed cloud computing process significantly reduces the training time by nearly half.

These experiments results confirm the importance of considering semantics in texts while performing topic classification task. In order to differentiate one topic from others, both symbolical words and semantical meanings are important. In order to classify documents into one topic, a good topic model needs to capture not only symbolical words but also their contexts, such as semantical meanings and semantical structures.

The proposed relation extraction algorithm was published in a conference paper entitled "An Entity-Based Algorithm for Multiple-Relation Extraction from Single Sentences" [Hao et al., 2017]. The proposed topic classification incorporating ontology knowledge with LDA was presented in a conference entitled "A self-Training Ontology-Driven Approach for Topic Classification (ST-OLDA)".

Organisation of the Thesis

The rest of the thesis is organised as follows. Chapter 2 presents a literature review about background techniques, including Natural Language Processing, existing relation extraction techniques, machine learning and classification techniques. Chapter 3 describes the topic model with ontology and the self-training process. This chapter also presents the experimental results of the combined self-training ontology-driven approach for topic classification. In this chapter, an ontology-driven topic model is proposed so that the semantical meanings of words can be considered when classifying documents based on topics. In order to reduce the required amount of pre-classified training data, a self-training procedure is introduced into the topic classification process. Next, an entity-based relation extraction algorithm is explained in Chapter 4. These extracted relation information are then combined with the ontology-driven topic model in Chapter 5, resulting in a novel relation-ontology driven topic classification technique. Chapter 5 also presents the distributed cloud computing process of

this topic classification technique. Finally, Chapter 6 concludes the whole thesis with some potential future work.

Chapter 2

Background

The topic of this thesis is interdisciplinary involving natural language data, computational linguistics, information extraction, modelling, classification and machine learning. Therefore, Natural Language Processing techniques for data processing are described. Besides, existing information extraction approaches for relation extraction from biomedical text data are also introduced. Furthermore, the related works for machine learning and classification of natural language documents are also covered for further study.

This chapter is divided into three main sections. Section 2.1 introduces some standard Natural Language Processing (NLP) techniques such as word segmentation, part-of-speech tagging and Named Entity Recognition. Some state-of-the-art techniques and applications of NLP tasks are also described in this section. Some state-of-the-art techniques and applications of NLP tasks are also described in this section. And Section 2.2 discusses four existing relation extraction approaches with corresponding examples. Section 2.3 presents two commonly used machine learning techniques for classification.

2.1 Introduction to Natural Language Processing

In order to process and analyse natural language text, Natural language processing techniques are used. Natural Language Processing (NLP) is a cross-disciplinary field in artificial intelligence and computational linguistics in which computers can interact with humans and understand natural human languages. Modern Natural Language Processing algorithms are improved by statistical machine learning. Without the direct manual coding of a large number of rules, machine learning enables computers to learn linguistic rules automatically through statistical analyses of large corpora from real-world human natural language. By attaching weights to each input feature, computers are able to make probabilistic decisions and predictions when trying to understand natural languages.

Due to the natural complexity and ambiguity, Natural Language Processing has a wide range of focused research areas such as Natural Language Generation (NLG), Natural Language Understanding (NLU), and natural language meaning extraction and summary. In this section, we first describe some standard techniques including *word segmentation*, *Part-of-Speech (PoS) tagging*, *parsing* and *Named Entity Recognition (NER)*. They are fundamental tasks when working with natural language texts. We also present the improvements of these NLP techniques when using word embedding with neural network and machine learning. Some state-of-the-art open-source platforms for NLP tasks are also described in this section.

Word Segmentation is an important Natural Language Processing problem. Due to the fact that words are often not specifically space-separated in English and other languages, it enables computers identifying and extracting valid terms like "San Francisco" from a continuous word stream. Traditional word segmentation systems employ supervised models containing complex sets of hand-written rules, decision trees, etc. These systems require a large amount of training data with correct annotations. In addition, supervised models are unable to solve new language problem whose linguistic properties were not covered by the computational learning model [Mochihashi et al., 2009]. For example, a model trained with an English dataset cannot deal with Chinese linguistic problems. Traditional unsupervised learning techniques such as neural networks [Pei et al., 2014] and genetic algorithms have contributed to automatic word segmentation in recent years. Furthermore, Bayesian networks are becoming the dominant approach nowadays because it provides an intuitive graphical visualisation of the probabilistic model and the conditional dependent. In addition, Bayesian Networks successfully represent the joint probability distribution so that the computational complexity of the inferences can be significantly reduced [Mochihashi et al., 2009]. In order to perform word segmentation in cross-lingual language models, machine learning techniques were utilised to increase the dictionary size or consider the content in historical documents [Liu and Wang, 2016, Homburg and Chiarcos, 2016, Kavitha et al., 2017]. Unlike words in English that can be easily recognised by the space token as a word divider, languages that do not have obvious word delimiters such as Chinese, Korean and Japanese do not have a clear word divider. By utilising deep learning techniques, Chinese Word Segmentation (CWS) is able to treat segmented words as basic units for operations. Each segmented word can be represented by a fixed-length vector, so that these representations of Chinese words are able to be processed by deep learning models in the same way as to how English words are processed [Yang et al., 2018, Li et al., 2019b]. Word segmentation is a fundamental NLP technique to solve NLP tasks such as sentiment mining [Shi et al., 2015], topic identification [Ehsan and Shakery, 2016], and language detection [Potrus et al., 2014].

Part-of-Speech (POS) tagging is a process of assigning labels to a word in a text based on both its definition and its context. In English, typical labels include noun, verb, adjectives, etc. *Parsing* is the process of analysing a string of words based on the rules of formal grammar and POS tags. When performing POS tagging and parsing to a sentence, a Penn Treebank style sentence with corresponding POS tagged words is produced to show its grammatical constructs. For example, Figure 2.1 shows the result of this process for the sentence. However, words are ambiguous due to the nature of human languages. For example, the word "increase" can be either a noun or a verb in different sentences. The tagging accuracy is restricted by the ambiguity of words. Hidden Markov Model (HMM) is one of the most widely used models for unsupervised inference in stochastic taggers [Banko and Moore, 2004, Collins, 2002, Lee et al., 2000, Thede and Harper, 1999]. Other models such as maximum entropy models [Ratnaparkhi et al., 1996], conditional Markov models [Klein and Manning, 2002, McCallum et al., 2000], conditional random fields [Lafferty et al., 2001], cyclic dependency networks [Toutanova et al., 2003], and Dynamic Bayesian Networks [Goldwater and Griffiths, 2007, Reynolds and Bilmes, 2005] are also helpful for part-of-speech tagging tasks. In part of speech (POS) tagging and parsing, the main challenge is to predict the right tags for both in-vocabulary (IV) and out-of-vocabulary (OOV) words. Recently, artificial neural networks, such as multi-layer perceptron (MLP) and long short term memory (LSTM), have been applied to POS tagging and parsing to overcome this challenge since they have high generality capabilities. Zhang et al. proposed an effective sequence-to-sequence neural model for Chinese word segmentation and POS tagging, based on a well-defined transition system, by using LSTM neural network structures [Zhang et al., 2018b]. By using well-trained character-level embedding, their neural joint model obtained the best-reported performances on five different datasets. [Yan et al., 2020] proposed a graph-based model to integrate Chinese word segmentation and dependency parsing, which is more concise with fewer efforts of feature engineering compared with transition system. They also combined their model with a character-level pre-trained language model to reduce the performance gap of parsing between joint models and gold-segmented word-based models[Yan et al., 2020]. Besharati et al. proposed an innovative model that combines a Hidden Markov Model and a single-layer bidirectional LSTM model to perform POS tagging and parsing in the Persian language. Their model successfully improved the accuracy compared against a simple second-order hidden Markov model HMM and a simple LSTM neural model. Same as word segmentation, part-of-Speech tagging and parsing in themselves may not be the solution to any particular NLP problem. It is, however, a pre-requisite process to simplify a lot of more complicated NLP tasks, such as text to speech conversion [Ning et al., 2019] and opinion extraction [Zhang et al., 2019].

"Female hormones lower magnesium but increase calcium levels which enhance migraine ubiquitousness." [Dhillon et al., 2011]

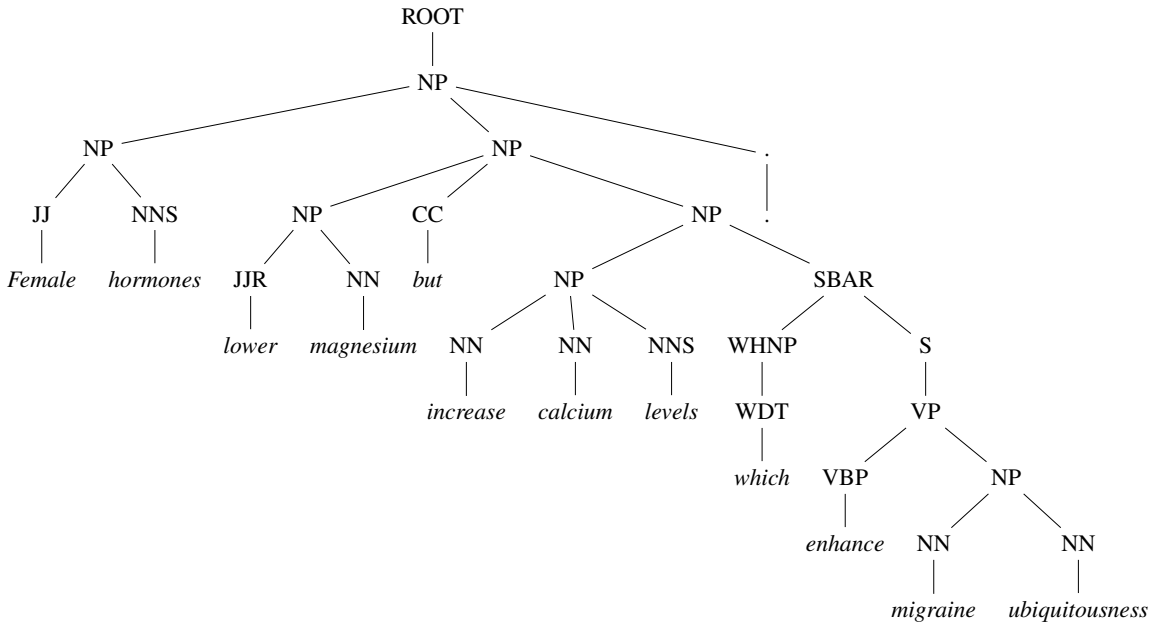


Fig. 2.1 Sample output result of the pre-processing of a sentence.

Named Entity Recognition (NER) is a cross-field task of Natural Language Processing and information extraction. It helps computers locate and classify named entities (NEs) that are rigid designates [Nadeau and Sekine, 2007] in natural texts into pre-defined categories such as generic NEs (e.g., person, organisations and location) and domain-specific NEs (e.g., certain biological species, substances proteins, enzymes, and genes) [Li et al., 2020]. The major difference between POS and NER is that the former focuses on grammatical roles whereas the latter focuses on named entities. Named Entity Recognition tasks can be divided into two major problems: the identification of names which is similar to the problem of word segmentation, and the classification of names into different types [Tjong Kim Sang and De Meulder, 2003]. Statistical models with linguistic grammar-based techniques are used for Named entity recognition systems. However, they require a large amount of manual effort to annotate training data. In 2011, Collobert et al. proposed neural network NER systems to minimise feature engineering efforts. Their models became popular because they typically do not require domain-specific resources like lexicons, and are thus poised to be more domain-independent. Since then, various neural architectures have been proposed, mostly based on deep learning models such as recurrent neural networks (RNN) over characters, sub-words and/or word embeddings. Yang et al. proposed a neural reranking model for

NER, where a convolutional layer with a fixed window-size is used on top of a character embedding layer. Their model leveraged recurrent neural network models to learn sentence-level patterns that involve named entity mentions [Yang et al., 2017]. Yadav et al. proposed an RNN model to learn specific representations of the prefixes and suffixes of words, which are then combined with the words or the character-level information to perform NER. Their approach achieved state-of-the-art results on the CoNLL 2002 Spanish and Dutch and CoNLL 2003 German NER datasets in multilingual and multi-domain [Yadav et al., 2018]. Jia et al. proposed a cross-domain Language Model as a bridge for NER domain adaptation, performing cross-domain and cross-task knowledge transfer by designing a novel parameter generation network with a bi-directional LSTM layer [Jia et al., 2019]. Xia et al. presents a novel framework for Multi-Grained Named Entity Recognition (MGNER). Unlike traditional NER approaches annotating entities consecutively, MGNER detects and recognises entities on multiple granularities: it is able to recognise named entities without explicitly assuming non-overlapping or totally nested structures [Xia et al., 2019]. NER is also an important pre-processing step for a variety of NLP applications such as information retrieval, question answering, machine translation, etc. There are some different scoring techniques for evaluating the quality of a named entity recognition system's output. F-measures [Tjong Kim Sang and De Meulder, 2003] is one of the most widely used methods for exact-match evaluation. With such measurements, a *true positive* (TP) represents an entity that has been correctly identified by the NER, a *false positive* (FP) represents an entity that has been incorrectly identified (i.e., it should not have been extracted); and a *false negative* (FN) represents an entity that should have been extracted but was missed by the NER. These definitions were introduced by CoNLL [Sang and De Meulder, 2003]. Using these definitions for true/false positives/negatives, precision and recall are defined in the usual manner as shown in Equation 2.1 and 2.2 [Perry et al., 1955]. The precision represents the percentage of the NER system results which are correctly recognised and the recall represents the percentage of total entities correctly recognised by the NER system [Nadeau and Sekine, 2007, Li et al., 2020].

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.2}$$

The F-measure of an approach is computed in the traditional balance F-score, as the harmonic mean of precision and recall [Perry et al., 1955]:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

With these fundamental techniques, NLP is being used to more sophisticated tasks such as question answering, sentiment analyses, speech recognition, information extraction (IE), statistical machine translation, and semantic web information extraction recent years [Wallis and Nelson, 2001]. Techniques utilising neural networks [Mikolov et al., 2013c], dimensionality reduction [Li et al., 2015] and probabilistic models [Globerson et al., 2007] with *word embedding* have been shown to boost the performance in NLP tasks involving word similarity and word analogy/analysis in deep learning studies [Socher et al., 2013a,b].

Word embedding is the collective name for a set of language modelling and feature learning techniques in NLP where words or phrases are represented as numerical vectors with certain distributions. Transitional methods such as bag-of-word (BoW) models have been proved unable to classify short texts [Sriram et al., 2010], as they ignore the order and semantic relations between words. Being a more advanced model, word embedding has been shown to preserve semantic types and syntactic relationships. It involves a mathematical mapping from space with many dimensions per word to a continuous vector space with a much lower dimension [Mikolov et al., 2013a]. The vectors of words in the same embedding space have similar semantic relationships, enabling the advancement of clustering tasks in NLP [Mikolov et al., 2013c].

The technique of representing words as space vectors was first introduced with the development of the vector space model (VSM) in the 1970s [Salton et al., 1975]. Then the introduction of latent semantic analysis (LSA) helps to reduce the number of dimensions by using singular value decomposition [Sahlgren, 2015]. Based on LSA, Blei et al. proposed an improved model based on the Dirichlet prior probability distribution [Blei et al., 2003] – the so-called Latent Dirichlet Allocation (LDA). In 2000, Bengio et al. proposed a series of *Neural probabilistic language models* that learns a distributed representation of words so that they can reduce the high dimensionality of words representations in contexts. Two different types of word embedding techniques are commonly used [Lavelli et al., 2004]: 1) words are represented as vectors of co-occurring words: this technique normally utilises bag-of-words and ignores the semantical context of the words; 2) words are represented as vectors of linguistic contexts in which the words occur: this technique is able to include the semantical context of the words in representation [Mnih and Hinton, 2009]. After 2010, word embedding techniques have been developed significantly because important

improvements had been made on the quality of vectors and the training speed of machine learning models. Google created *word2vec* in 2013, which is an open toolkit to train vector space models for computing vector representations of words [Mikolov et al., 2013b]. Pennington et al. proposed a global log-bilinear regression model called Global Vectors for Word Representation(GloVe) [Pennington et al., 2014]. It is an open-source unsupervised learning algorithm for obtaining vector representations for words, which is trained on aggregated global word-word co-occurrence statistics from a corpus. These representations can then be subsequently used in many natural language processing applications and for further research.

Recently, new word embedding techniques relying on neural network architectures and machine learning frameworks have achieved an improved performance, resulting in a much broader application area.

Two most popular deep learning frameworks among researchers for NLP tasks are described here. *Tensorflow* is a free and widely adopted machine learning platform for fast numerical computing [Abadi et al., 2016]. It was created and released by Google under the Apache 2.0 open source license. It is powerful to build easy models and robust models on any platform. It provides simplified APIs which can be used both in research and development and in production systems. *PyTorch* is an open-source machine learning library used for various types of applications such as computer vision and natural language processing [Rao and McMahan, 2019]. It was developed by Facebook's AI Research lab (FAIR) [Patel, 2018]. Google, Facebook, Microsoft and many other organisations across industries are increasingly using these two frameworks as the foundation for their most important machine learning (ML) research to solve NLP tasks.

In 2018, a team from Google AI Language proposed a Bidirectional Encoder Representations from Transformers (BERT), which is pre-trained on a large corpus comprising the Toronto Book Corpus and Wikipedia [Devlin et al., 2018]. Language representation models analyse a string of text either from left to right or combined left-to-right and right-to-left training during word embedding [Peters et al., 2018, Radford et al., 2018]. Instead of reading the text input sequentially (left-to-right or right-to-left), BERT looks at the entire sequence of words at once using a procedure called "masked LS" (MLM). With MLM, BERT is able to obtain contextual representations of words using both left and right contexts, resulting in bidirectional language representations. BERT also employed a Next Sentence Prediction (NSP) which uses pairs of sentences as input to predict if the second sentence in the pair is the subsequent sentence in the original text. BERT can be applied to different NLP tasks in a pretty straight forward way by simply adding one additional output layer to the core model. This pre-trained representation model reduces the required efforts for heavily engineered

tasks to construct individual-specific architectures. To the best of our knowledge, BERT is the first pre-trained representation that achieves state-of-the-art results on different tasks without modifying the architecture for specific tasks.

GPT and GPT-2 were proposed by OpenAI as unidirectional language models trained by Generative Pre-Training method [Radford et al., 2018, 2019]. Such unidirectional language models enable them to predict the next token in a sequence based on all of the previous context. GPT was pre-trained on the Toronto Book Corpus with a causal language modelling (CLM) objective, resulting in a left-to-right architecture. GPT-2 improved GPT by employing 1.5 billion parameters and a much larger dataset consisting of 8 million web pages. As a result, GPT-2 is able to be applied and achieves good results in many tasks across many domains. Both GPT and GPT-2 have been proven to significantly improve upon the state-of-the-art in various NLP tasks such as reading comprehension, text summarisation, translation and question answering. At May 2020, OpenAI proposed the third generation language prediction model GPT-3 [Brown et al., 2020]. Unlike the previous two versions, GPT-3 is an autoregressive language model that automatically generates human-like text. It was pre-trained with 175 billion parameters, making it the largest non-sparse language model to date. Therefore, GPT-3 achieves strong performance on many NLP tasks from different domains. GPT-3 is able to produce highly coherent text that is difficult to distinguish from that written by a human, which brings both benefits and risks [Sagar, 2020].

Because GPT-3 can "generate news articles which human evaluators have difficulty distinguishing from articles written by humans,"[4] GPT-3 has the "potential to advance both the beneficial and harmful applications of language models."[1]:34 In their May 28, 2020 paper, the researchers described in detail the potential "harmful effects of GPT-3"[4] which include "misinformation, spam, phishing, abuse of legal and governmental processes, fraudulent academic essay writing and social engineering pretexting".[1] The authors draw attention to these dangers to call for research on risk mitigation.[1]:34

Facebook proposed two methods to learn cross-lingual language models (XLMs): one unsupervised method to learn cross-lingual representations that only relies on monolingual data, and one supervised that utilises parallel data that existing improves cross-lingual pre-training [Lample and Conneau, 2019]. Their methods significantly outperform the existing state-of-the-art on cross-lingual classification, unsupervised machine translation and supervised machine translation.

In this work, we utilised existing toolkits from StanfordNLP [Manning et al., 2014b] and OpenNLP [Morton et al., 2005] to perform some fundamental NLP tasks. StanfordNLP is an open-source Python package to perform natural language analysis. It contains tools, which can be used in a pipeline, to perform common NLP tasks such as word segmentation,

POS tagging and NER. It is designed to be parallel among more than 70 languages, using the Universal Dependencies formalism [Berzak et al., 2016]. StanfordNLP is normally treated as a state-of-the-art toolkit when performing standard NLP tasks. This package is built with highly accurate neural network components that enable efficient training and evaluation with various datasets from different domains. The OpenNLP is another commonly used machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as sentence and word segmentation, POS tagging and NER. It contains components which enable others to execute the respective NLP task, to train a new model for specific NLP task and to evaluate the performance of a model. They are all accessible via its application program interface (API) and a command line interface (CLI).

2.2 Relation Extraction

A substantial amount of valuable information is recorded in the form of unstructured text data, such as news, emails, journal articles and academic papers. Identifying entities, relations and information of interest is the pre-requisite of extracting structured knowledge from these unstructured raw texts, which has received growing interest recently [Zweigenbaum et al., 2007, Kreimeyer et al., 2017, Wang et al., 2017]. A relationship extraction (RE) task is a subtask of information extraction (IE), which requires the detection and classification of semantic relationship mentions within a set of text or XML documents. The biomedical literature is one body of knowledge of this form. The use and management of biomedical relations and information to stay updated and informed are important issues [Singh and Gupta, 2017]. Biomedical relation extraction outputs play important roles in question-answering systems [Aggarwal and Zhai, 2012], diagnosis categorisation [Luo et al., 2017] and clinical decision support [Chowdhury and Mahbub, 2013]. However, Biomedical literature often contains special named entities that are long and complicated. They also tend to use long and complicated sentences, which is more difficult to analyse automatically compared to texts from general domains. In this section, we first describe some state-of-the-art work on relation extraction on various datasets from general domains, such as news articles, web pages, etc. Then we focus on four types of approaches for relation extraction on biomedical datasets.

A typical relation extraction model approach this task by extracting a list of triples from the text, i.e., $REL(e_1, e_2)$, which represents the relation REL exists between entity e_1 and entity e_2. Existing models can be divided into two major categories: the pipelined approach, which first uses NER models to identify entities, and then uses relation extraction models to identify the relation between each entity pair; and the joint approach, which combines the NER model and the relation model through different strategies.

For the pipelined approaches, relation extraction is tackled by using two separate models for entity recognition and relation classification. The introduction of neural networks constitutes the state-of-the-art, such as RNNs [Zhang and Wang, 2015, Zhang et al., 2018c], recursive neural networks [Socher et al., 2012] and CNNs [Zeng et al., 2014]. Pre-trained Transformer models also have been used for relation classification [Verga et al., 2018, Wang et al., 2019]: The input text is fed through a Transformer model, such as BERT and GPT-2, and the resulting embeddings are classified as relations. However, these approaches with neural models have difficulties to capture all the lexical, semantic and syntactic cues [Li et al., 2019c], especially when (1) entities are far away (i.e. relations are embedded in complex sentence structures such as clauses structures or conjunction structures); (2) one entity is involved in multiple triplets (i.e. text contains one to many, many to one or many to many relations); or (3) relation spans have overlaps (i.e. text containing entities e_1, e_2, e_3, e_4 has relations $REL(e_1, e_3)$ and $REL(e_2, e_4)$). As we shall see, our proposed algorithm described in Chapter 4 can successfully solve these drawbacks by including several extensions.

In recent years, there has been an increasing interest of performing joint entity-relation extraction, recognising entities and relations at the same time from unstructured texts data. Miwa and Sasaki proposed the first joint entity-relation extraction model. They treat the joint entity-relation extraction task as a table-filling problem, where each cell of the table corresponds to an entity pair of the sentence. The table is filled with relations by minimising a scoring function based on several standard NLP features such as POS tags and entity labels [Miwa and Sasaki, 2014]. Vu et al. took this idea to solve the task, whereas they employed a bidirectional recurrent neural network to label each relation pair [Vu et al., 2016]. Miwa and Bansal proposed a stacked model for to perform joint entity-relation extraction task. They employed a bidirectional sequential LSTM model to tag entities of interest and a bidirectional tree-structured RNN to label relations based on the dependency parse tree [Miwa and Bansal, 2016]. Zhou et al. combined a bidirectional LSTM and a CNN to extract a high-level feature representation of the input text. Their model extracts a fewer number of relations compared to the table-filling approaches because they only extract entities for the most likely relations [Zhou et al., 2017]. Bekoulis et al. proposed a model that employs a Conditional Random Fields (CRF) layer for entity recognition. Their model treats the relation extraction task as a multi-head selection problem so that it can potentially identify multiple relations for each entity Bekoulis et al. [2018]. Their model does not require any manually extracted features or the use of any external NLP tool. Transformer networks use their idea to approach joint entity-relation extraction as a multi-head self-attention problem, resulting in an improvement in this task. Li et al. tackle the joint entity-relation extraction task by incorporating a BERT-based multi-turn question answering model [Li et al., 2019c]. Answers

to the manually defined question templates constitute extracted entities and their relations. The main limitation of their approach is that these hand-crafted question templates require domain expertise (e.g., for biomedical or clinical corpora). Similarly, Giorgi et al. employed the pre-trained, transformer-based language model BERT to performs entity recognition and relation extraction simultaneously without relying on external NLP tools such as dependency parsers [Giorgi et al., 2019]. Eberts and Ulges also proposed an attention-based Transformer type network with BERT embedding [Eberts and Ulges, 2019]. To the best of our knowledge, their approach achieved the best performance of relation extraction on CoNLL04 dataset (The F1 score for relation extraction is 71.47 and for entity extraction is 88.94) [1]. As we shall see, our proposed entity-based algorithm outperforms their algorithm. Besides all the advantages of joint entity-relation extraction models, problems such as co-reference resolution and relation extraction synchronisation remain existing challenges [Ghamami and Keyvanpour, 2018].

The rest of this section mainly focuses on existing approaches for relation extraction task in biomedical text data. Different approaches and relation extraction systems have been developed and utilised to extract biomedical concepts and relationships from text, including MedLEE [Friedman et al., 1994], KnowledgeMap [Denny et al., 2003], cTAKES [Savova et al., 2010], HiTEX [Goryachev et al., 2006], and MedTagger [Liu et al., 2013]. Existing approaches for biomedical relation extraction can be divided into four categories: approaches based on co-occurrence, link-based approaches (pattern-based), machine learning approaches and rule-based approaches. In what follows, we briefly introduce these approaches giving a typical example for each. For the purpose of evaluating the relation extraction algorithms by means of F-measures, a *true positive* (TP) represents a relation that has been correctly identified by the extraction algorithm, a *false positive* (FP) represents a relation that has been incorrectly identified (i.e., it should not have been extracted); and a *false negative* (FN) represents a relation that should have been extracted but was missed by the extraction algorithm. Using these definitions for true/false positives/negatives, precision and recall are define as in Equation 2.1, 2.2 and 2.3. The first three of these techniques can only deal with simple relations between two entities connected by a target word and generally achieve relatively low precision and recall. Applying them in different domains can be time-consuming. Rule-based extraction normally achieves higher precision and can be applied in a variety of domains [Sharma et al., 2010]. In this work, we proposed a novel rule-based method for relation extraction, which can be applied in both the biomedical domain and general domain. [2]

[1] See https://paperswithcode.com/sota/relation-extraction-on-conll04

[2] This technique is discussed in more detail in Chapter 4

2.2.1 Co-occurrence Approaches

Co-occurrence approaches provide the simplest way to detect relations when the two entities frequently co-occur within a collection of texts or sentences [Garten et al., 2010]. They assume two entities are related based on lexical statistics such as word frequency counts. In 1996, Gordon and Lindsay developed the computer-based statistical methods to discover the connection between Raynaud's disease and dietary fish oil through medical literature [Gordon and Lindsay, 1996]. They considered both the frequency of tokens within text data and the number of records containing various tokens to indicate literature relations and potential entity-relation discoveries. Srinivasan developed open and closed algorithms to identify and rank key terms based on term-weighting strategies [Srinivasan, 2004]. Hristovski et al. presented BITOLA to discover novel relations between a given gene candidate of interest and other concepts from literature [Hristovski et al., 2005]. They include background knowledge about different diseases and genes from resources like LocusLink and the Human Genome Organization (HUGO). Torvik and Smalheiser proposed a two-node search interface to discover a certain number of B-terms according to eight strongly correlated complementary features [Torvik and Smalheiser, 2007]. They estimated the overall number of relevance between B-terms and a given two-node search using a logistic regression model. Their system simplified the process of a two-node search and was capable of applying in various general domains. He et al. designed a web-based tool for Protein-Protein interactions (PPI) finding based on co-occurrences and interaction words from PubMed abstracts [He et al., 2009]. They considered shared evidence from human PPI databases and Gene Ontology (GO) database. Senger et al. created a one-step solution called *prolific* (protein-literature investigation for interacting compounds) to extract protein names or sequences information using frequencies of co-occurrences. They automatically extracted up to 69% drug-protein relationships [Senger et al., 2012].

However, co-occurrence approaches have some drawbacks. Since it is likely that the two entities might be mentioned together without any relation, Zweigenbaum et al. considered frequency-based scoring schemes to eliminate such relation [Zweigenbaum et al., 2007]. The schemes give higher scores to the relation when it is more unlikely to be observed. Co-occurrence approaches result in high recalls. But they may have poor precisions because biomedical texts contain complex sentences embedding multiple entities, most of which are not always actually related. For example, *"Low magnesium intakes and blood levels have been associated with type 2 diabetes, metabolic syndrome, elevated C reactive protein, hypertension, atherosclerotic vascular disease, sudden cardiac death, osteoporosis, migraine headache, asthma, and colon cancer"* [Rosanoff et al., 2012] contains 12 entities but the last ten entities are not related to each other. To address this, filtering steps were introduced to

improve the precision of these systems. Lindsay and Gordon generated candidate intermediates and targets by lexical statistics alone and then culled these lists with two human filters to eliminate candidates based not only on general knowledge but also on nonspecialist medical knowledge [Lindsay and Gordon, 1999]. Kabiljo et al. removed sentences that do not match lexico-syntactic criteria before extracting candidate relations and proteins were considered related only when a relation word was located between them [Kabiljo et al., 2009]. Another issue is that co-occurrence approaches normally cannot identify the semantical context of the relationships. Therefore, the extracted relation data still require manual annotation before employing. Furthermore, co-occurrence methods are robust since they do not require any linguistic analysis. They are usually used for comparing against other more advanced methods as a baseline method [Pyysalo et al., 2008, Bunescu et al., 2006, Henry and McInnes, 2017].

Example

We explain the approach proposed by Senger et al. in detail because they achieved better results compared to others based on co-occurrences. They created a *prolific* system to provide a one-step solution for closing the gap between protein information in literature and sequence information on proteins. Their database was composed of 11.7 million PubMed abstracts, 35.4 million PubChem [Bolton et al., 2008] compounds with 606.3 million (partially overlapping) synonyms. They connected all the compounds with their 28.3 million original characters and 2.0 million synonyms with their UniProtKB/Swiss-Prot [Magrane and Consortium, 2011] protein IDs. They also attached Gene Ontology Annotation (GOA) terms [Barrell et al., 2009] and their "gene symbols" to protein IDs. They also generated a "stop word list", consisting of words with a high frequency of occurrence for unspecific meanings, such as "ANOVA" used in statistical context but also an abbreviation for "RNA-binding protein Nova-2". They annotated abstracts from a local database with protein IDs and protein synonyms by Whatizit web services [Rebholz-Schuhmann et al., 2008] and parsed the annotated texts after filtering with the protein "stop word list" to extract "protein-article" relationships. Then they searched in all PubMed abstracts for compound synonyms based on the synonym processing rules [Hettne et al., 2009] to extract "article-compound" relationships after filtering with the compound "stop word list". At last, the extracted "protein-article-compound" relationships were categorised into four types:

1) Co-occurrence of protein and compound in the abstract. For example, the paper with PubMed ID 23935933 is about a compound *glucocorticoid receptor modulator* and a protein *Hsp70 gene promoter*. This abstract contains co-occurrence of protein and compound.

2) Co-occurrence in the same sentence. For example, the sentence *The lack of a Compound A-induced increase in Hsp70 protein levels in A549 cells is not mediated by a rapid proteasomal degradation of Hsp70 or by a Compound A-induced general block on translation,* contains co-occurrence of a compound *Compound A-induced* and a protein *Hsp70*.

3) Co-occurrence in the same sentence enclosing a functional process or molecular function. For example, the sentence *Recently, we developed N-[4-[6-(isopropylamino) pyrimidin-4-yl]-1,3-thiazol-2-yl]-4-[11C]methoxy-N-methyl-benzamide ([11C]ITMM) as a useful positron emission tomography (PET) probe for mGluR1 in clinical studies,* contains co-occurrence of a molecular function *N-[4-[6-(isopropylamino)pyrimidin-4-yl]-1,3-thiazol-2-yl]-4-[11C] methoxy-N-methyl-benzamide* and a protein *mGluR1*.

4) Co-occurrence in a sentence enclosing curated relationship verbs. For example, *Metabotropic glutamate receptor subtype 1 (mGluR1) is a crucial target in the development of new medications to treat central nervous system (CNS) disorders*, contains co-occurrence of a protein *Metabotropic glutamate receptor subtype 1* and a disease *central nervous system (CNS) disorders* enclosing a verb *is*.

Those extracted relationships were assembled and stored de-normalised in a NoSQL database, allowing fast access to a query started with any names, IDs or sequences.

2.2.2 Pattern-based Approaches

Pattern-based approaches extend co-occurrence approaches by identifying the relations if the two entities often co-occur with a common term across a collection of corpus. They normally rely on a set of patterns for relation extraction. According to diverse ways to generate patterns, these approaches can be categorised into supervised approaches and unsupervised approaches.

Supervised pattern-based approaches require a set of corpus created by domain experts, which are time-consuming [Hakenberg, 2010]. They also can be affected by various factors such as the irregularity in biomedical entity names (e.g. TP53, FtsZ), abbreviations (e.g. VD, FOXP3), or punctuation (e.g. 4-dihydroxyphenylalanine, AY-27). These approaches differ from each other as the process varies from one corpus to another, which makes it difficult to evaluate and compare due to corpus incompatibilities. Blaschke and Valencia developed one of the first systems to extract phrases expressing protein-protein interactions using hand-crafted regular expressions [Blaschke and Valencia, 2002]. Zhou et al. started with simple patterns such as *protein1-relation-protein2* [Zhou et al., 2008]. They extracted

a limited number of relations between proteins connected by a set of pre-defined relation words such as *inhibit, bind, activate,* etc. Their system yields high precision, but the recall rate remains low. Syntactic analysis or semantic parsing of sentence structures, such as POS tagging and parsing, has been introduced to define grammars before extracting relations. Such procedures are referred to as surface-pattern approaches, and they improved the performances of supervised pattern-based approaches while worked not well on complex sentences [Hao et al., 2005]. Overall, supervised pattern-based approaches result in high precisions, but low recalls. They can be applied to different knowledge domains after carefully fixed to some specific problems. However, there is still no guarantee that they can extract relations that do not occur in the pre-defined training patterns after tuning. Until recently, methods based on manually generated patterns still could not achieve satisfactory results.

Unsupervised pattern-based approaches were proposed to increase the recall of supervised pattern-based approaches. Only input data is given to the system, and there are no correct answers nor trainers. Algorithms are left to their own abilities to discover and generate patterns in the sentences. Hao et al. designed a minimum description length (MDL)-based pattern-optimisation algorithm to reduce and merge patterns, which significantly increased generalisation power, and hence the recall and precision rates [Hao et al., 2005]. In general, two techniques for automatically generating patterns were developed depending on whether they require a corpus or not. Approaches for generating patterns without a corpus normally use bootstrapping techniques [Wang et al., 2011]. They obtained patterns from the input seeds, such as a small list of PPI pairs, and extracted new relations of the same types as the generated patterns from unstructured texts. They repeated this process until no more new patterns can be found. However, these approaches may result in large sets of noisy patterns [Rebholz-Schuhmann et al., 2010, Fox et al., 2010]. Nguyen et al. proposed several filters for simple pattern selections to improve precision of relations extraction at a slight drop of the recall [Nguyen et al., 2010]. Hakenberg et al. developed a filter by ranking sentences with relevance containing novel interactions or evidence for physical interactions to rid of noisy patterns [Hakenberg et al., 2010]. On the other hand, generating patterns directly from corpora can also help to eliminate noisy patterns. Yakushiji et al. developed a method of automatically constructing patterns on predicate-argument structures (PASs) obtained by full parsing from a smaller training corpus [Yakushiji et al., 2006]. Both unsupervised pattern-based techniques depend more on linguistic features than manually supervised approaches, resulting in better recall rates. Le Minh et al. employed heuristic rules and dictionaries to annotate event trigger words and event extraction was based on patterns created from dependent graph [Le Minh et al., 2011].

However, it remains challenging to choose the right amount of patterns for relation extraction because some of the automatically generated patterns may overmatch texts due to their generic while some of them can not match unseen texts due to their specific. Choi proposed a novel tree pattern expression (TPE) to represent various structural patterns and reduce pattern-matching complexity significantly [Choi, 2011].

Unsupervised pattern-based approaches achieve better performances in generating patterns than supervised approaches, resulting in better performances in relations extraction while the noisy patterns cause reduction of the precisions. As for approaches based on co-occurrence, they also do not employ any NLP techniques for linguistic analysis so that they are not able to consider some important aspects when extracting relations such as the semantic context of relations.

Examples

To the best of our knowledge, approach proposed by Hao et al. was the first unsupervised pattern-based approach for relation extraction in biomedical literature. Therefore, we explain this approach in detail. For a set of sentences $S = s_1, S-2, \ldots, s_n$, they aim to extract as et of interactions $I = I_1, I_2, \ldots, I_m$. They designed a pattern set function to manually generate patterns for relation extraction [Hao et al., 2005]. The pattern set P^* is the P which minimises the expected risk $R(P)$:

$$P^* = \arg\min_P R(P) = \arg\min_P \int_S L(S,P)\,\mathrm{d}G(S) \tag{2.4}$$

where $G(S)$ is the probability distribution of S, and $L(S,P) = |I^* - F(S,P)|$ is the loss function and I_* is the true interactions set defined by S. They also introduced the MDL principle [Rissanen, 1978] to solve the trade-off problem between generalisation power and accuracy:

$$M_{mdl} = \arg\min_M K(M) + K(D|M) \tag{2.5}$$

where $K()$ is Kolmogorov complexity, D represents the data using model M. They assumed the interaction set I to be a sequence given by $I = I_1 I_2 \ldots$ and defined $K(I) = K(P) + K(I|P)$ as the description length of I through P, where $K(P)$ is the description length of pattern set P and $K(I|P)$ is that of I given P. They also considered the Hamming distance of two interaction sequences:

$$d(I, I*) = \sum^{n_i=1} \delta(I_i, I_i^*), \tag{2.6}$$

where $\delta(I_i, I*_i) = \begin{cases} 1 & I_i \neq I_i^* \\ 0 & I_i = I_i^* \end{cases} \ldots$

Therefore their optimized pattern was defined as:

$$P^* = \arg\min_P K(I) + \log_2 d(I, I^*) = \arg\min_P K(P) + K(I|P) + \log_2 d(I, I^*) \tag{2.7}$$

where I and I^* are the extracted and optimal interaction sequences, respectively, and $d(I, I^*)$ is the number of differences between I and I^*.

Based on the pattern set P, they manually generated 192 patterns based on 963 sentences and 1435 interactions. They implemented a cross-validation experiment and achieved a precision of 0.851 and a recall of 0.558 with 30 patterns.

2.2.3 Machine Learning Approaches

Machine learning approaches label and segment sentences to extract relations automatically with annotated corpora on the biomedical domain. They are typically modelled as a classification problem with the help of NLP tools to pre-process unstructured sentences. Many machine learning approaches employed various general models such as Hidden Markov Model [Collier et al., 2000], Conditional Random Fields(CRF), Naïve Bayes classifier [Gildea and Jurafsky, 2002] and Support Vector Machine (SVM). In general, machine learning approaches can be categorised into feature-based methods and kernel-based methods based on the nature of the input to the classifier.

Feature-based Approaches

Feature-based approaches use lexical, syntactic and semantic features to represent the data characteristics for deciding whether the entities in a sentence are related or not. They normally are specific to binary relations between two entities (e.g. protein-protein or gene-protein) using POS tags and dependency parse trees. Feature-based approaches can be classified into shallow (partial) parsing based approaches and deep (full) parsing based approaches based on the complexity of the employed features. The former approaches explore syntactic information from only a part of the sentences to improve the efficiency and reliability, while the latter approaches analyse the whole sentence structure, resulting in the better performance but increased computational complexity. Gupta et al. proposed a hybrid semi-supervised approach that combines parsed sentences with grammatical structures for extracting simple relations from free-text mammography reports. However, the number of features increases significantly to thousands when more feature types are included, whilst

decreasing the precision of the system [Van Landeghem et al., 2010]. Our entity-based approach can solve more general features without increasing computing time. Zheng et al. presented an effective model of a parsed sentence to extract relations by introducing vectors of context to represent all labelled nodes adjacent and nonadjacent to it to capture the direct and indirect substructures' information. However, unlike our entity-based approach can be applied in different domains, their method only considered the distance between context vectors to detect drug-drug interactions (DDIs). In order to make full use of each model and avoid their individual weaknesses, Abacha et al. tried to combine models with different features together to improve the performance for relation extraction [Abacha et al., 2015]. However, the combination of models requires more computational resources to train the classifiers compared to our entity-based algorithms. Miwa et al. designed a rich feature vector, including bag-of-words (BOW) features, shortest path features and graph features, and aggregated them into a single support vector machine modified with corpus weighting (SVM-CW) to complete the task of multiple corpora PPI extraction [Miwa et al., 2009b]. BOW features consist of words (lemma forms) that appear before, between and after the entities of interest. The shortest path features consist of syntactic information extracted from the shortest walk on the target pair entities which are represented by two nodes of a parse tree. Graph features consist of all the nonzero labels and weights from parse structure sub-graphs and linear order sub-graphs from the dependency parser. These features were widely used for machine learning approaches afterwards, achieving better performances among all other machine learning approaches for relation extraction problem. Based on these common features, various relation extraction systems have been proposed [Sætre et al., 2007, Van Landeghem et al., 2008, Kim, 2008, Kim et al., 2008, Ahmed et al., 2009, Niu et al., 2010]. However, feature engineering is a precise and costly task [Aggarwal and Zhai, 2012]. Due to the complexity and high cost faced with locality and bias to features in relation extraction task, feature-based methods have a decreased accuracy and generality when dealing with large scale data and dimension [Yin et al., 2017].

Example of Feature-based Machine Learning Approaches

We explain the method proposed by Niu et al. in detail because it achieved better results compared to other feature-based machine learning approaches. They proposed an interaction detection method using various features to automatic identify PPIs in text [Niu et al., 2010]. They considered a vector of features, including context, lexical forms and positions within a sentence that contains two target proteins *M1 and M2*. They used three types of features to describe the position of the two proteins: their indices in the sentence, their distance between each other and the number of other proteins between them. They also considered lexical

forms in the sentence to extract context information about those extracted relations: 3 tokens on the left of M1, 3 tokens on the right of M2 and all tokens between the two target proteins. Their set of keywords features was developed based on the list from Plake et al. [Plake et al., 2005]: 1) the lexical forms of keywords, 2) the position of keywords, 3) the distance between keywords and the target protein nearest to it. They also incorporated some patterns from Plake et al. [Plake et al., 2005] for pattern-matching, such as *proteinA* | ***form/forms*** ...***complex with*** ...| *proteinB* (the bold words have to be matched exactly). In addition, they considered the phrase and word dependency in a sentence as a feature to achieve better performances.

For example, the sentence *The lack of a Compound A-induced increase in Hsp70 protein levels in A549 cells is not mediated by a rapid proteasomal degradation of Hsp70 or by a Compound A-induced general block on translation* contains two target protein candidates *Compound A-induced* and *Hsp70*. Distance between each other is two tokens, and there are no other proteins between them. The algorithm also considered *lack of a*, *increase in* and *protein levels in* based on the lexical forms. The identified keyword is *increase*, which is located between the target proteins and the token distance between the keyword and the nearest target protein is 0. The protein *Hsp70* actually have dependent words *protein levels*, indicating the target protein name is *Hsp70 protein levels*. Therefore, the extracted relation from the example sentence is Compound A-induced | increase | Hsp70 protein levels.

Their algorithm achieved an average precision of 0.60 and recall of 0.30, whereas our entity-based approach performs better. [3]

Kernel-based Approaches

Kernel-based approaches use various kernels to compute the similarity between two instances based on the similarities of their representations. Unlike feature-based approaches, these approaches can make better use of the structural representations of entities such as syntactic parse trees and dependency graphs. Ahmed et al. proposed the first kernel-based approach for PPIs extraction using string-kernels to quantify the number of subsequences that are common to both strings for similarity calculation [Ahmed et al., 2009]. Bunescu et al. introduced Bag-of-words (BOW) kernel to calculate the similarity between two feature vectors, consisting of unsorted sets of words [Bunescu et al., 2005]. Giuliano et al. described the shallow linguistic (SL) kernel, consisting of two kernels: the global kernel and local context kernel [Giuliano et al., 2006]. The former kernel count common words in three feature set of BOW vectors obtained from two sentences, while the latter kernel contains linguistic features generated from words appear before and after the two entities. Moschitti

[3] The detail of the evaluation experiments of our entity-based approach is described in Section 4.6

provided a simple algorithm using sub tree (ST) kernel to compute the similarity between two input syntactic trees based on the number of common sub-trees [Moschitti, 2006]. Bunescu and Mooney proposed the first subsequence kernel using three types of subsequence patterns that are typically employed in natural language to assert relationships between two entities [Bunescu and Mooney, 2005a]. Kim et al. proposed a walk-weighted subsequence kernel considering not only of non-contiguous syntactic structures but also of semantic and lexical structures [Kim et al., 2010]. The combination of various structural analysis enables the system to learn more valuable aspects from a rather small amount of training data. Bunescu and Mooney considered the shortest path between two entities in the same sentence in the dependency graph to extract a relation between them [Bunescu and Mooney, 2005b]. Airola et al. designed a graph kernel to calculate the similarity between the dependency structure graph and linear order graph by counting weighted shared paths of all possible paths [Airola et al., 2008a,b]. In order to make full use of each kernel and avoid their individual weaknesses, researchers tried to combine them to improve the performance for relation extraction [Miwa et al., 2008, Li et al., 2008, Miwa et al., 2009a]. However, the combination of kernels requires more computational resources to train the classifiers [Fayruzov et al., 2009]. Kernel-based methods work well in dealing with transferring implicit data to vector space. However, they do not consider any semantic information [Murugesan et al., 2017].

Example of Kernel-based Machine Learning Approaches

We explain the method proposed by Segura-Bedmar et al. in detail because it achieved better results compared to other kernel-based machine learning approaches. They proposed a machine learning-based method using shallow linguistic kernel for drug-drug interaction (DDI) extraction in biomedical texts [Segura-Bedmar et al., 2011]. They created the first annotated corpus DrugDDI corpus using DrugBank database [Wishart et al., 2008]. They treated the DDI extraction task as a drug pair classification task. Therefore, they generated datasets to train and test the classifier from the DrugDDI corpus by enumerating all possible ordered pairs of sentence entities: $\{(D_i, D_j) : D_i, D_j \in D, 1 \leq i, j \leq N, i \neq j, i < j\}$, where S stands for the sentence, N stands for the number of drugs and D stands for the set of drugs. The example was labelled 0 if the interaction did not exist between the two candidate drugs. Otherwise, it was labelled 1. Since they did not consider the order of the drugs in the sentence, the number of examples was $C_{N,2} = \begin{pmatrix} n \\ 2 \end{pmatrix}$. A kernel function is defined as a binary function $K : X \times X \to [0, \infty)$ that maps a pair of instances $x, y \in X$ to their similarity

score $K(x,y)$. The kernel function must satisfy the following:

$$\forall x,y \in X : K(x,y) = \langle \phi(x), \phi(y) \rangle \tag{2.8}$$

where $\phi : X \rightarrow F \subseteq R^n$ is a mapping from the input space X to a vector space F. The normalization equation was defined as:

$$\forall x,y \in X : K(x,y) = \langle \phi(x), \phi(y) \rangle = \sum_{i=1}^{m} \phi_i(x) \times \phi_i(y). \tag{2.9}$$

$$K(x_i,x_j) = \frac{\langle \phi(x_i), \phi(x_j) \rangle}{\| \phi(x_i) \| \| \phi(x_j) \|} \tag{2.10}$$

Based on the above equations, they proposed the shallow linguistic kernel (K_{SL}) as the linear combination of a global context kernel (K_{GC}) and local context kernel (K_{LC}):

$$K_{SL}(R_i,R_j) = K_GC(R_i,R_j) + K_LC(R_i,R_j) \tag{2.11}$$

where the global context kernel and the local context kernel were defined as:

$$K_{GC}(R_1,R_2) = K_FB(R_1,R_2) + K_B(R_1,R_2) + K_BA(R_1,R_2) \tag{2.12}$$

$$K_{LC}(R_1,R_2) = K_{\text{left}}(R_1,R_2) + K_{\text{right}}(R_1,R_2) \tag{2.13}$$

After a few experiments for investigating the suitable window-size of the local context and n-gram of the global context for better performances, they selected the model maximising both the F-measure and the precision (n-gram =3, window-size = 3) so that they could avoid overloading database curators with too many false positives during the process. In the end, they achieved a precision of 0.5103 and recall of 0.7282, whereas our entity-based approach performs better. [4]

2.2.4 Rule-based Approaches

Rule-based approaches use NLP techniques and templates generated manually by domain experts or learned automatically from training data to identify semantic entities and extract associations connected by some specific verbs [Fundel et al., 2006, Mykowiecka et al., 2009, Song et al., 2015, Kim et al., 2017]. They extend the pattern-based approaches by adding constraints to express more general patterns such as determining the sentiment of relations [Fox et al., 2010, Koike et al., 2005, Kim et al., 2007]. Their templates tend to be

[4] The detail of the evaluation experiments of our entity-based approach is described in Section 4.6

too specific, focusing on specific verbs between two entities while our entity-based approach is focusing on semantic verbs and relying solely on the existence of multiple entities within a sentence. In addition, by expressing rules with a set of procedures or heuristic algorithms instead of specific constraints, these approaches improved the performances of relation extraction [Rinaldi et al., 2007, Fundel et al., 2007, Rinaldi et al., 2006]. Unlike pattern-based approaches, standard NLP techniques such as POS tagging, parsing, and NER are used to analyse text data before relation extraction. Some simple co-occur relation structures, such as *Entity-Verb-Entity*, *gene product acts as a modifier of gene*, were considered for relation extraction at first [Proux et al., 2000]. Ono et al. manually generated a set of rules based on syntactic features to extract negative relations such as *Protein1-not verb-Protein2* from complex sentences [Ono et al., 2001]. Sharma et al. proposed a algorithm to extract single relations from simple sentences [Sharma et al., 2010]. However, their algorithm mainly focused on five types of entities: food, disease, protein, chemical and gene. Raja et al. implemented a web-based text mining tool called PPInterFinder to extract human PPIs from biomedical literature by applying a set of manually defined rules on grammatically parsed sentences and matching the syntactic structure of the sentence with a dictionary of patterns [Raja et al., 2013]. Cohen et al. employed the OpenDMAP semantic parser with manually-written rules to detect trigger words in the training data to extract events, arguments, negations and speculations [Cohen et al., 2009]. Song et al. defined rules that rely mainly on syntactic deep parsing and manually specified dictionaries to extract relations represented as a pair of entities, linked by a directed arc [Song et al., 2015]. Kim et al. used an association rule learning algorithm to obtain relationships [Kim et al., 2017]. They utilised these measures to extract meaningful relationships and weights for relationships.

As in machine learning approaches, manually generated rules are limited by expensive time-consuming and domain constraints. It is also not realistic to cover all the possible descriptions of relations in texts. Therefore, researchers have been trying to automatically generate rules for relation extraction from literature. Phuong et al. generated rules automatically using a set of sample sentences parsed by a link grammar parser [Phuong et al., 2003]. In order to remove the non-protein interactions, they incorporated heuristic rules based on morphological clues and domain-specific knowledge. Dynamic programming was employed to automatic learn PPI rules based on POS tags [Huang et al., 2004]. Thomas et al. incorporated the grammatical information encoded in the types of the dependencies in dependency trees (DTs) instead of only exploiting topological features [Thomas et al., 2011]. A large set of linguistic rules was inferred using information about interacting proteins alone, which were then fine-tuned based on shallow linguistic features and the semantics of dependency types.

In general, rule-based approaches achieve better performance compared to pattern-based and machine learning approaches since their rules are at rather abstract levels such as syntactic structures, grammatical or semantic structures. Therefore, rule-based approaches perform better when applied to new domains with a small amount of training data. However, the recall rates of these approaches remain low since the pre-defined rules can only deal with obvious cases.

Example

We explain the method proposed by Raja et al. in detail because it achieved better results compared to other rule-based approaches. They implemented PPInterFinder to extract human PPIs from biomedical literature using rule-based approaches [Raja et al., 2013]. They developed a vast relation keywords dictionary consisting of 354 relation words and categorised them into 88 subtypes by identifying the common root word. Based on the relation keywords dictionary, they designed three abstract forms considering the position of the proteins and the keywords. For example, the expression *PROTEIN1 interacts with PROTEIN2* conforms with the first form definition *PROTEIN1 - token* - RELATION - token* - PROTEIN2*. Table 2.1

Table 2.1 Abstract forms for PPI candidate pair

Form 1:	PROTEIN1 - token* - RELATION - token* - PROTEIN2
Form 2:	RELATION - token* - PROTEIN1 - token* - PROTEIN2
Form 3:	PROTEIN1 - token* - PROTEIN2 - token* - RELATION

In addition to these abstract forms, they also set seven rules for identification of candidate PPI pairs: 1) the position of relation keyword with proteins, 2) the number of tokens/words between the protein pairs, 3) simple sentences with two proteins and a relation keyword, 4) simple sentences with two proteins, a relation keyword and a negation keyword 5) complex sentences having more than two proteins and a relation keyword, 6) complex sentences having more than two proteins, a relation keyword and a negation keyword, 7) complex sentences having more than two proteins and two negation keywords.

According to the three abstract forms and the seven rules, they defined 11 patterns using Tregex syntax [Levy and Andrew, 2006], such as *S ((NP≪PROTEIN1) $++ (VP≪RELATION) $++ (NP≪PROTEIN2))*.

To better understand this approach, we used the sentence *The lack of a Compound A-induced increase in Hsp70 protein levels in A549 cells is not mediated by a rapid proteasomal degradation of Hsp70 or by a Compound A-induced general block on translation* as an example. This sentence contains the abstract form 1, conducting as *Compound A-induced*

increase in Hsp70. The position of the only one relation keyword *increase* is located between the two proteins in this simple sentence. Therefore, one PPI candidate pair *compound A-induced* | *Hsp70* can be extracted from this sentence based on pattern 1.

They evaluated their PPI systems with five standard corpora: AIMED [Bunescu et al., 2005], BioInfer [Pyysalo et al., 2007], HPRD50 [Fundel et al., 2007], IEPA [Ding et al., 2002] and LLL [Nédellec, 2005]. They achieved an average precision of 0.85 and recall of 0.65, whereas our entity-based approach performs better. Our approach also can be easily applied in various domains with different corpus rather than only focus on PPIs.

2.3 General Introduction to Machine Learning

Machine learning (ML) is an application of Artificial Intelligence (AI) that studies algorithms and statistical models. The primary aim of ML is to enable computer systems to automatically perform a specific task and learn and improve from experience without being explicitly programmed or need human intervention, depending on patterns and inference instead [Bishop, 2006]. Computer programs start the machine learning process with getting access to data and observing data, such as examples, direct experience, or instruction. In order to look for patterns in data, machine learning algorithms build computational statistics models based on "training data". These models are then able to make decisions or predictions in the future. Machine learning algorithms can be used in a wide variety of applications, such as computer vision [Sebe et al., 2005, Grys et al., 2017], data mining [Michalski et al., 1998, Ivezić et al., 2019], email filtering [Alurkar et al., 2019, Mallampati et al., 2019] and *text classification* [Khan et al., 2010, Kadhim, 2019], etc.

Machine learning enables machines to perform automatic analysis of massive amounts of data. Even though it generally delivers faster, more accurate results when identifying profitable opportunities or dangerous risks, it may still require additional time and resources to achieve efficient training. Combining machine learning with cloud computing and cognitive technologies can make it even more effective in processing large volumes of information.

Machine learning algorithms are often categorised into four groups. Table 2.2 shows a general comparison of these four groups.

Table 2.2 Comparison of supervised machine learning and unsupervised machine learning

Features	Supervised Learning	Unsupervised Learning	Semi-supervised Learning	Reinforcement Learning
Type of problems	Regression and classification	Clustering and associations	Classification and clustering	Reward-based
Type of data	Labelled data	Unlabelled data	Partially labelled data	No predefined data
Training	External supervision	No supervision	Partially supervision	No supervision
Approach	Maps the labelled inputs to the known outputs	Understands patterns and discovers the output	Utilised approaches from both supervised and unsupervised	Follows the trial-and-error method
Computational Complexity	Computationally Simple	Computationally complicated	In between of supervised and unsupervised	Computationally complicated
Accuracy	Highly accurate	Less accurate	In between of supervised and unsupervised	Depends on the quality of environment

- *Supervised machine learning* can apply what has been learned in the past to new data using labelled training data to predict future events [Mohri et al., 2018]. By analysing the labelled training dataset, the supervised learning algorithm produces an inferred function that maps an input object (typically a vector) to the desired output value based on example input-output pairs [Russell and Norvig, 2016]. The algorithm is able to provide predictions of labels for any unseen input after sufficient training. It can also calculate the errors between its output and the correct, intended output so that the model can be modified accordingly. Compared to the other three machine learning methods, supervised machine learning often achieves higher accuracy with a relatively low computational cost. Supervised machine learning algorithms can be used to solve problems like regression (the output variable is a real value) and classification (the output variable is a category). These are explained in the following sections. Some of the most widely used supervised learning algorithms are: Support Vector Machines (SVM) [Cortes and Vapnik, 1995], linear regression [Freedman, 2009], logistic regression [Tolles and Meurer, 2016], naive Bayes [Rish et al., 2001], etc.

- *Unsupervised machine learning* is a type of self-organised Hebbian learning that can find previously unknown patterns in a training dataset that is neither classified nor labelled. Unsupervised learning studies how models can infer a function to describe a hidden structure from unlabelled data [Hinton et al., 1999]. Without knowing the correct output, the algorithm explores the training data and makes inferred functions to describe hidden structures and patterns from unlabelled data. Unsupervised machine learning algorithms can be used to solve a variety of problems such as clustering (to discover the inherent groupings in the data) [Kassambara, 2017] and associations (to discover rules that describe large portions of your data) [Lud and Widmer, 2000]. Even though unsupervised machine learning is usually computationally complicated and less accurate, it can help to find previously unknown patterns and discover unknown outputs in a dataset without pre-existing labels. Some of the most widely used unsupervised learning algorithms are: k-means [Garbade, 2018], Expectation–maximization algorithm (EM) [Dempster et al., 1977], Neural Networks [Sarle, 1994], etc.

- *Semi-supervised machine learning algorithms* fell between unsupervised learning (without any labelled training data) and supervised (with completely labelled training data) since they use a large amount of unlabelled data in conjunction with a small amount of labelled data. The systems that use this method have considerably improved learning accuracy [Zhu, 2005]. Usually, semi-supervised learning is chosen when the acquisition of the acquired labelled data requires skilled human agents, relevant

resources or physical experiments. This process can be time-consuming, expensive and infeasible, whereas acquiring unlabelled data is relatively inexpensive since it generally does not require additional resources. In such situations, semi-supervised learning can be of great practical value.

- *Reinforcement machine learning algorithms* interacts with software agents' environment by taking actions and discovers punishments or rewards. Reinforcement learning differs from supervised learning in two ways: It does not need labelled input and output pairs and does not need sub-optimal actions to be explicitly corrected. Instead, it focuses on finding a balance between exploration of uncharted territory and the exploitation of current knowledge [Kaelbling et al., 1996]. It is trained by trial and error search to allow machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximise its performance. Delayed reward feedback, known as the reinforcement signal, is required for the agent to learn which action is best.

In this work, we focus on categorisation tasks, specifically classification tasks. Categorisation plays an important role in machine learning, which aims to solve objects classification, recognition, differentiation and understanding [Cohen and Lefebvre, 2005]. In a categorisation task, objects are grouped into categories based on certain purposes, such as relationships of ideas or meanings of the categories and objects. Categorisation can be applied in many fields, such as natural language prediction, inference, decision processes [Frey et al., 2011].

Categorisation tasks can be differentiated into two types based on whether it is a supervised or unsupervised training procedure:

- *Classification* is the problem of assigning categories to new observations based on a set of training data in which category labels are provided to the learner for certain objects. It involves extracting information from the labelled data sets to achieve accurate prediction of class labels of unlabelled new data. This may require the abstraction of a rule or concept relating observed object features to category labels [Kotsiantis et al., 2007]. For example, deciding a given email as a "spam" email or "non-spam" email. And giving a diagnosis to a patient based on their characteristics (sex, blood pressure, presence or absence of certain symptoms, etc.). And in our work, assigning a document to a given topic based on the context. The algorithm that implements classification is known as a *classifier*, which is a classification algorithm consisting of mathematical functions, that maps input data to a class [Cooke, 2011]. In machine learning, the observations are often known as instances, the properties of observations are treated as explanatory variables, and the predicted categories are known as outcomes, which are

considered to be possible values of the dependent variable. The explanatory variables are termed features, and the possible categories to be predicted are classes.

- *Clustering* is the task of grouping objects by similarity into classes so that objects in the same cluster are more similar to each other than to those in other clusters. It involves recognising inherent structure in a data set in which no labels are supplied and generating a classification structure [Kaufman, 2012].

The procedure for building a supervised model to solve a classification problem is as follows. It begins with separating available data into a training set and a testing set. Then a strategy is required to find the architecture of the model that performs classification [Swain and Sarangi, 2013]. The strategy (or algorithm) is defined by specifying an optimisation procedure and a mathematical form of the problem. Once the model is found, it is treated as a classifier that is trained through observing the training data set and assigning objects into certain classes. The model then compares observations by means of similarities and adjusts parameters of the algorithm.

In each observation, the data set can be treated as a set of quantifiable properties, which technically are referred to as variables or features. The vectors contain multiple numbers that represent the values of the axis in different dimensions. The space location that stores these numbers corresponds to the variables or features.

The ability of a classifier adapting to the unknown future data is the most important factor once the model is built and trained. This is done by examining the model with the testing data set. The result highly depends on the characteristics and the representation of the data. The procedure for measuring the results is called evaluation. Measuring precision and recall are popular approaches [Tjong Kim Sang and De Meulder, 2003], while simply performing an accuracy test is the most common way to evaluate a model [Rossi et al., 2003].

Naturally, an advanced classifier such as Here, we describe a typical supervised learning classifier, support vector machines (SVMs [Cortes and Vapnik, 1995]), which are scalable and efficient to classify multiple topic categories for huge text corpus [Chen, 2018]. In our work, a typical C-SVM was used to perform the *text classification* task. This is the same choice for other existing works [Allahyari and Kochut, 2015, Liu et al., 2019]. The classification results can be used to evaluate the accuracy of different topic model approaches.

2.3.1 Support Vector Machine

In machine learning, SVMs [Cortes and Vapnik, 1995] are supervised learning models for analysing data in order to achieve classification and regression analysis. Given a set of training data, each labelled with one or the other of two categories, an SVM training algorithm

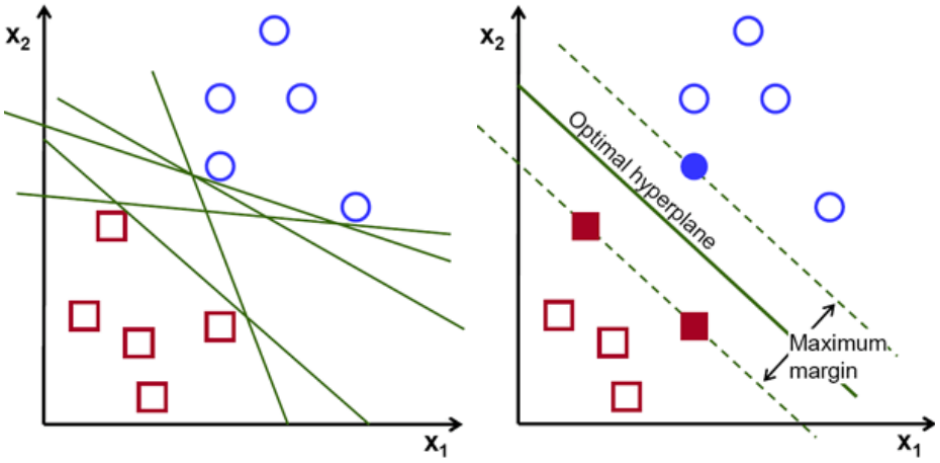


Fig. 2.2 The geometric building of an optimal hyperplane for two-dimensional input space

assigns new unlabelled data to one category or the other, making it a non-probabilistic binary linear classifier [Khamar, 2013]. An SVM model is a representation of the data as points mapped in space, data of the separate categories are divided by a clear gap (or hyperplane) that is as wide as possible [Lilleberg et al., 2015]. New unlabelled data are then represented by the same space and assigned to a category according to which side of the gap they fall on. Beside linear classification, SVM can also be to perform a non-linear classification by including the kernel trick, so that the inputs can be mapped into high-dimensional feature spaces [Boser et al., 1992].

For a typical SVM, given a training dataset $(x_i, y_i)_{i=1}^{n}$, where y_i are either 1 or −1, each indicating the topic category to which the point x_i belongs. SVM algorithm assumes that the topic categories are linearly separable. The maximum-margin hyperplane that used to separate categories is defined in Equation 2.14, where $\mathbf{w}$ is an adjustable weight vector and b is a bias. The desired hyperplane ensures that the distance margin ρ between the hyperplane and the nearest point x_i from either category is maximised. This desired hyperplane is the optimal hyperplane. Figure 2.2 shows the geometric building of an optimal hyperplane for two-dimensional input space. Each data point must lie on the correct side of the hyperplane, as defined in Equation 2.15.

$$\mathbf{w}^T \cdot x + b = 0 \tag{2.14}$$

$$y_i(\mathbf{w} \cdot x_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n \tag{2.15}$$

To extend SVM to cases in which the data are not linearly separable, the hinge loss function is introduced and defined in Equation 2.16. This function is zero when the constraint in Equation 2.15 is satisfied. When Equation 2.15 is not satisfied, in other words, for a data point on the wrong side of the hyperplane, the hinge loss function's value is proportional to the distance from the hyperplane. In this case, Equation 2.17 should be minimised. Here λ is the trade-off between increasing the margin size and ensuring that the x_i fall on the correct side of the hyperplane.

$$max(0, 1 - y_i(\mathbf{w} \cdot x_i - b)) \tag{2.16}$$

$$[\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(\mathbf{w} \cdot x_i - b))] + \lambda \|w\|^2 \tag{2.17}$$

Computing the SVM classifier, i.e., minimising Equation 2.17, can be done by solving a constrained optimization problem with a differentiable objective function as follows. For each $i \in 1, \ldots, n$, a variable $\zeta_i = max(0, 1 - y_i(\mathbf{w} \cdot x_i - b))$ is introduced, which is the smallest nonnegative number satisfying $y_i(\mathbf{w} \cdot x_i - b) \geq 1 - \zeta_i$. Thus Equation 2.17 can be rewritten as in Equation 2.18, which is called the *primal problem* [Boser et al., 1992].

$$\begin{aligned} \text{minimise } & \frac{1}{n}\sum_{i=1}^{n} \zeta_i + \lambda \|\mathbf{w}\|^2 \\ \text{subject to } & y_i(\mathbf{w} \cdot x_i - b) \geq 1 - \zeta_i \\ \text{and } & \zeta_i \geq 0, \text{ for all } i = 1, \ldots, n \end{aligned} \tag{2.18}$$

In this work, we focus on a C-Support Vector Classification model (C-SVM) from LIBSVM proposed by Chang and Lin [Chang and Lin, 2011] to solve *text classification* problem. After representing documents using different topic models to construct a training dataset, they are fed into the C-SVM to obtain a classifier. The primal problem of such classifier can be written as Equation 2.19.

$$\begin{aligned} \text{minimise } & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda \sum_{i=1}^{n} \xi_i \\ \text{subject to } & y_i(\mathbf{w}^T \phi(x_i) + b) \geq 1 - \xi_i, \\ \text{and } & \xi_i \geq 0, \text{ for all } i = 1, \ldots, n \end{aligned} \tag{2.19}$$

where $\phi(x_i)$ maps x_i into a higher-dimensional space. Due to the possible high dimensionality of the vector variable $\mathbf{w}$, they solve a dual problem instead following Equation 2.20. Here $\mathbf{e} = [1, \ldots, 1]^T$ is the vector of all ones, Q is an $n \times n$ positive semidefinite matrix, $Q_{ij} \equiv$

$y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function.

$$\begin{aligned} \text{minimise } & \frac{1}{2}\alpha^T Q\alpha - \mathbf{e}^T\alpha \\ \text{subject to } & \mathbf{y}^T\alpha = 0, \\ \text{and } 0 \leq \alpha_i \leq \lambda, & \text{ for all } i = 1, \ldots, n \end{aligned} \tag{2.20}$$

After Equation 2.20 is solved, they utilised the primal-dual relationship to obtain the optimal **w** following Equation 2.21 and the decision function following Equation.

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{2.21}$$

$$sgn(\mathbf{w}^T \phi(x) + b) = sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, \mathbf{x}) + b) \tag{2.22}$$

Finally, the classifier is constructed and trained successfully and ready to make predictions on any testing dataset using $y_i \alpha_i \forall i$, weight vector **w**, bias b, label names[5], and other information such as kernel parameters.

[5]In LIBSVM, label names are mapped to ± 1 by assigning the first training instance to $y_i = +1$

Chapter 3

Ontology-Driven Approach for Topic Classification

The first task of a *topic classification* system is to develop a *topic model* to summarise and represent unstructured texts written in natural language. The obtained *topic model* can then be combined with a classifier to perform *text classification* task. In the chapter, we aim to address **(RQ1)** by introducing an ontology-driven topic classification method with LDA topic model.

Topic modelling techniques can summarise texts into topics and *topic classification* techniques identify topic terms and classify texts accordingly. Latent Dirichlet Allocation (LDA) is one of the most commonly used *topic modelling* techniques [Li et al., 2016, Hsu and Chiu, 2017, Burkhardt and Kramer, 2018]. LDA employs a probabilistic model that projects a document into a topic space matrix using the Dirichlet probability distribution [Girolami and Kabán, 2003]. Each topic is represented by a collection of words and their probability distribution [Blei et al., 2003]. An LDA model can be generated and trained by either supervised and unsupervised machine learning techniques. In general, LDA models produced by supervised techniques vastly outperform those produced by unsupervised techniques [Li et al., 2016, Hsu and Chiu, 2017, Burkhardt and Kramer, 2018]. However, supervised techniques need to be trained by a manually generated and classified dataset, which is very costly to produce [Ko and Seo, 2009]. And as a result, these training datasets are usually small. While larger training datasets not only ensure better generalisation, they also provide better accuracy. Wang et al. incorporated expert knowledge when generating the topic model so that it does not need large training datasets. However, this still requires a large amount of human effort [Pavlinek and Podgorelec, 2017]. In order to overcome the cost of obtaining a large pre-classified training dataset, Ocepek et al. suggested introducing a self-training phase to automatically enlarge an initially small amount of training data annotated by humans [Ocepek

et al., 2015]. Pavlinek and Podgorelec combined this self-training phase with LDA resulting in a technique called *Self-Training LDA* (ST-LDA) [Pavlinek and Podgorelec, 2017]. Once the enlarged training dataset is generated, the topic classifier can then be performed using a conventional supervised technique, such as Support Vector Machine (SVM).

Conventional LDA approaches always use words as self-contained tokens so that they normally ignore the fact that words may have multiple meanings and that different words may have the same meaning, which result in limited performances of the topic models they produce [Campbell et al., 2015b]. For example, the sentences

"Google is launching their new phone."

and

"Microsoft is stepping into the study of advanced electronics."

may not be classified into the same category when represented by an LDA topic model because they do not have any relevant words in common. However, these sentences *are* related because Microsoft and Google are both "companies" involved with "technology". In this work, we aim to bridge this gap by including some semantical concepts associated with the words "Microsoft" and "Google". This can be done by making use of a database containing a good amount of cross-domain ontological knowledge such as ConceptNet and DBpedia.

ConceptNet is a freely-available semantic network that contains the meanings of words in natural language and the common-sense relationships between them [Speer et al., 2017]. The multilingual knowledge in ConceptNet is collected from a variety of resources, including crowd-sourced resources (such as Wiktionary [1] and Open Mind Common Sense [Speer et al., 2008]), and expert-created resources (such as WordNet [University, 2010] and JMDict [Breen, 2004]). ConceptNet is a commonly used resource for researches working with ontology knowledge, such as sentiment analysis [Chauhan and Meena, 2020, Bandari and Bulusu, 2020] and question answering [Talmor et al., 2018, Basu et al., 2020]. DBpedia can also provide structured information about over 6.0 million entities associated with a set of concepts describing its general properties within the ontology. These entities and their concepts together construct a consistent ontology.

Our approach utilises these semantical concepts from either ConceptNet or DBpedia to include implicit relationships between words into topic models and therefore increase the overall accuracy of the classification. In our previous example, the words "Microsoft" and "Google" would be associated through the concepts "company" and "technology" that they

[1] https://www.wiktionary.org/

share in ConceptNet. Including this ontological knowledge as an intermediate component in LDA has the following advantages: (i) it allows the topics to be defined more generally in terms of ontological concepts rather than specific words so that they can capture the semantical meaning of the words more accurately; (ii) as a side-effect, we will see that this extra component helps to reduce the time required to construct a topic model. In virtue of the use of this ontological knowledge, we call the resulting technique *Ontology-Driven Latent Dirichlet Allocation* (OLDA).

As for LDA, OLDA can also employ a self-training phase in order to reduce the amount of human effort by enlarging the initial amount of manually classified data. The inclusion of the self-training phase enables OLDA to deal with small corpus. Accordingly, we call the variant using the self-training phase *Self-Training Ontology-Driven Latent Dirichlet Allocation* (ST-OLDA). The self-training can be performed with any appropriate procedure. Two alternatives were considered in this work: a relatively ad hoc method employing a logistic regression model; and the procedure proposed by Pavlinek and Podgorelec [Pavlinek and Podgorelec, 2017], which employs Gibbs sampling. The former is faster to train, but its classification is less accurate. In our experiments, the combination of Pavlinek's self-training technique with OLDA outperformed it with LDA [Pavlinek and Podgorelec, 2017] by as much as 11.01% (in the R52 dataset), which confirms the advancement of including the ontology component. Comparing against state-of-the-art knowledge-based approach [Allahyari and Kochut, 2015], the inclusion of logistic regression increases the accuracy of the classification regardless of the self-training method employed by between 3 and 7 percentual points (depending on different datasets). Comparing against state-of-the-art word embedding based approaches [Fu et al., 2016, Liu et al., 2019], our knowledge-based OLDA approach does not rely on the performances of external word embeddings even though ours achieves lower accuracy of the classification. This work has been accepted and presented in the 2019 International Conference on Computer Science and Information Technology entitled as *"A Self-Training Ontology-Driven Approach for Topic Classification (ST-OLDA)"*.

The remainder of this chapter is organised as follows. Section 3.1 provides background information about logistic regression model and some topic modelling techniques. Section **??** presents our new OLDA approach. Section 3.3 describes the two self-training techniques and how they can be incorporated with OLDA. Section 3.4 describes the results of our experimental analysis and Section 3.5 summarises with a discussion.

3.1 Background

As the approach introduced here employs logistic regression, this section introduces the logistic regression model. It also introduces existing topic modelling and classification techniques against which our approach is compared.

3.1.1 Logistic Regression Model

In statistics, the logistic model is used to compute the probability of a specific class or event happening, such as pass or fail, win or lose, alive or dead and healthy or sick. In machine learning, it can be extended to solve multi-class tasks, such as determining whether an image contains a specific object. The detected object would be assigned a probability between 0 and 1, and the sum of these probabilities is one.

A logistic regression model uses a logistic function to estimate and model a binary dependent variable, although many more complex extensions exist [Tolles and Meurer, 2016]. In statistics, a binary logistic model has two dependent variables where each can be two possible values, such as pass or fail are assigned with 0 and 1, which are indicator variables. The binary logistic regression model can be extended to more than two dependent variables: *multinomial logistic regression* can model categorical outputs with more than two values, and ordinal logistic regression can model multiple ordered categorical outputs [Walker and Duncan, 1967b].

The logistic function takes any real input $t, t \in \mathbb{R}$, and outputs a value between 0 and 1 [Hosmer Jr et al., 2013]. A standard logistic function $\sigma : \mathbb{R} \to (0,1)$ is as:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}} \tag{3.1}$$

Assuming that t is a linear function of a single explanatory variable x, the input t can then be expressed as:

$$t = \beta_0 + \beta_1 x \tag{3.2}$$

where β_0 is the intercept from the linear regression equation and β_1 is the regression coefficient.

And the general logistic function $p : \mathbb{R} \to (0,1)$ can now be written as in equation 3.3.

$$p(x) = \sigma(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{3.3}$$

3.1.2 Topic Modelling and Classification

Topic modelling is a type of statistical modelling to summarise a collection of documents into the abstract "topics". Intuitively speaking, given that a document is about a particular topic, certain words would be expected to appear in the document more frequently, and others less frequently. A topic model captures this intuition using a mathematical technique, which analyses a set of documents to produce "topics" based on collections of related words and their statistics. The topic model is able to discover what the topics might be and what each document's balance of topics is. Topic modelling has been widely applied to various text mining tasks such as text classification [Hingmire and Chakraborti, 2014, Li et al., 2018, 2019a], word sense disambiguation [Hu et al., 2014, Kim et al., 2020], sentiment analysis [Yono et al., 2019, Lazaridou et al., 2013], and others. Manually assigning topics to documents are prone to subjectivity and not able to scale up, especially when dealing with a massive number of data [Mei et al., 2006, Wang and McCallum, 2006].

Vector Space Model (VSM) representation is a common topic modelling approach, where topics are based on words as independent units with mathematical weights computed based on Term-Frequency-Inverse Document Frequency (TF-IDF) [Salton and Buckley, 1988]. However, only considering word frequency without any context is not sufficient to differentiate between topics in many situations because the order of the appearance of the words may result in various meanings and topics [Sriurai, 2011]. To address this, Nigam et al. employed a Naïve Bayes (NB) classifier to perform parameter estimation task of a statistical Expectation Maximisation (EM) model (EM-NB) [Nigam et al., 2000]. Nigam et al. used Naïve Bayes classifier to estimate parameters from pre-classified instances and then to assign probabilistically weights to unclassified instances and assign their classes accordingly. In a second iteration, these new classified instances are considered to re-estimate and adjust parameters, and so it iterates until the results converge. A significant drawback of this approach is that it converges to a local optimum. Deerwester et al. proposed a Latent Semantic Analysis (LSA) approach to interpret documents as latent concepts in a low dimensional semantic space [Deerwester et al., 1990]. Hofmann further enhanced LSA so that topics are represented by multinomial word distributions using the probability density function, yielding a technique called Probabilistic Latent Semantic Analysis (PLSA) [Hofmann, 1999]. However, PLSA is prone to incorrectly classify documents which cause overfitting problems. In order to address these problems, Blei et al. proposed an improved model based on the Dirichlet prior probability distribution [Blei et al., 2003] – the so-called Latent Dirichlet Allocation (LDA). In LDA, each document is represented as a multinomial distribution of topics, where each topic is represented as a distribution of words.

$$\begin{pmatrix} d_{11}, d_{12}, \ldots, d_{1k} \\ d_{21}, d_{22}, \ldots, d_{2k} \\ \vdots \\ d_{n1}, d_{n2}, \ldots, d_{nk} \end{pmatrix} = \begin{pmatrix} q_{11}, q_{12}, \ldots, q_{1l} \\ q_{21}, q_{22}, \ldots, q_{2l} \\ \vdots \\ q_{n1}, q_{n2}, \ldots, q_{nl} \end{pmatrix} \times \begin{pmatrix} p_{11}, p_{12}, \ldots, p_{1k} \\ p_{21}, p_{22}, \ldots, p_{2k} \\ \vdots \\ p_{l1}, p_{l2}, \ldots, p_{lk} \end{pmatrix}$$

Δ (document/words) $\quad$ Θ (documents/topics) $\quad$ Φ (topics/words)

Fig. 3.1 *A typical schematic of LDA matrices*

Fig. 3.1 shows a typical schematic representation of LDA matrices. Here, $\mathscr{D} = \{D_1, \ldots, D_n\}$ is a collection of documents to be classified into the topics $\mathscr{T} = \{T_1, \ldots, T_l\}$ and $\mathscr{W} = \{W_1, \ldots, W_k\}$ is a set of words occurring in $\mathscr{D}$. Δ is the matrix associating documents to words, where each cell d_{ij} is 1 if the word $W_j \in \mathscr{W}$ appears in the document $D_i \in \mathscr{D}$ and 0 otherwise. LDA treats each document in a collection as having been created from several latent topics, each of which having an associated probability distribution of co-occurring words [Campbell et al., 2015b]. This multinomial probability distribution is captured by a $l \times k$ matrix Φ with the probabilities p_{ab} of each topic T_a ($1 \leq a \leq l$) being described by word W_b ($1 \leq b \leq k$). The LDA model aims to obtain from Δ and Φ a $n \times l$ documents/topics matrix Θ with the probabilities q_{xy} of each document D_x ($1 \leq x \leq n$) being associated with topic T_y ($1 \leq y \leq l$).

An LDA model is trained to obtain the topics/words matrix Θ and documents/topics matrix Φ in an unsupervised manner. One of the most widely-used training techniques is Gibbs sampling [Griffiths and Steyvers, 2004]. For the initial iteration, Gibbs sampling starts by randomly assigning probabilities to p_{ab_0} in Φ_0, and then the algorithm repeats over each word W_b of each document D_i in the training datasets for a number of iterations. For each iteration t, it samples a new probability p_{ab_t} computed by the conditional distribution of the word W_{b_t} given all other current topics/words probabilities $p_{a'b'_t}$ ($1 \leq a' \leq k$ and $a' \neq a$, $1 \leq b' \leq k$ and $b' \neq b$). The iteration process stops when the algorithm reaches a steady-state, resulting in a topics/words matrix Φ with obtained topics/words probability distributions. Finally, the desired documents/topics matrix Θ can be computed following Fig. 3.1.

Shortcomings of LDA

In spite of its strengths, LDA sometimes fails to capture the true semantical meanings of the topics due to problems caused by word-assignment ambiguity, homonyms and polysemous words [Ramage et al., 2009]. Some variations of the LDA model have attempted to reduce these noise and address these problems. Panichella et al. proposed a genetic algorithm to fine-tune the prior probabilities in Φ and Θ [Panichella et al., 2013]. Hsu and Chiu proposed

a supervised hybrid LDA approach using a genetic algorithm to optimise the weight vector of the documents-topics matrix Θ [Hsu and Chiu, 2017]. Krasnashchok and Jouili employed a named entity recognition technique to recognise and include domain-specific terms into a new weighted LDA model. This additional knowledge can improve interpretability, specificity and the diversity of the extracted topics [Krasnashchok and Jouili, 2018]. Hida et al. proposed a dynamic and static topic model (DSTM) for LDA to simultaneously consider the dynamic structures of the temporal topic evolution and the static structures of the topic hierarchy [Hida et al., 2018]. However, none of these approaches seems to address LDA's intrinsic inability to capture semantical meanings of words. Guo and Diab attempted to achieve a better understanding of word semantical meanings by exploiting dictionary definitions explicitly from WordNet [Guo and Diab, 2011], but WordNet ontologies are too fine-grained, resulting in a topic model of less generalisation power. Hulpus et al. incorporated structured knowledge from DBpedia with LDA. For each topic, they first found the terms with the highest marginal probabilities by LDA and then generated a set of ontological concepts from DBpedia to represent those terms of the topic. These concepts can then be used to construct a graph so that graph centrality algorithms can identify the most representative concepts for the topic [Hulpus et al., 2013]. However, their works basically treat topics as a multinomial distribution over words, which can not interpret the semantic of each topic in an accurate way. Recently, distributed word representations significantly improved the performances of topic modelling tasks. Fu et al. proposed a Word-Topic Mixture (WTM) model to obtain an improved word embedding representation and a topic model. They introduced an initial external word embedding into the Topical Word Embeddings (TWE) model [Liu et al., 2015] to learn word embeddings. Then probability distribution of vectors-word embeddings from TWE is integrated into the LDA by defining the probability distribution of topic models according to the idea of latent feature model with LDA (LFLDA) [Fu et al., 2016]. They achieved good results in 20 Newsgroups dataset: 80.94% classification accuracy. Similarly, Liu et al. incorporated word embedding and part-of-speech in their LDA topic model in order to capture the context of the words in documents [Liu et al., 2019]. Indeed their approach produced impressive results: their topic model with an SVM classifier achieved an average of 83.05% accuracy on the 20Newsgroups dataset. However, their approach requires a lot of manually classified training data. As we shall see, the performance of our proposed OLDA and ROLDA are comparable with [Fu et al., 2016] and [Liu et al., 2019], whereas ours do not rely on the performance of external word embeddings and is able to deal with small corpus. Allahyari and Kochut introduced another latent variable called *concept* into LDA between topics and words. Unlike LDA, their model treats each topic as a multinomial distribution over concepts and each concept as a multinomial distribution over

words [Allahyari and Kochut, 2015]. To the best of our knowledge, their OntoLDA model is the first model to introduce an intermediate concept variable into LDA. We take the same idea of the intermediate concept variable into LDA, resulting OLDA. Our work differentiates from OntoLDA in that we employed a logistic regression model to predict probability distributions of documents/topics and topics/concepts. In comparison, they employed conventional Gibbs Sampling for probability inference and training. As we shall see, OntoLDA's performance is lower than the accuracy of our proposed OLDA model. Employing other machine learning techniques such as LSTM may achieve better results on such a task. However, we chose logistic regression due to its simplicity and efficiency. As we shall see, even with a simple logistic regression model, the inclusion of the intermediate knowledge-based concept variable increased the classification accuracy as well as the construction speed of a topic model.

A secondary issue with LDA is that the training process of the topic model in a purely unsupervised manner results in difficulties to achieve an accurate topic classification. Researchers attempted to create semi-supervised LDA models. Wang et al. proposed a semi-supervised LDA model (ssLDA) by manually incorporating available expert knowledge [Wang et al., 2012], but this still requires a lot of human intervention. Wu et al. represented documents as concept vectors instead of word vectors using heuristic selection rules to select only related keywords rather than the full-text obtained from Wikipedia [Wu et al., 2017]. Gu et al. combined a supervised Bi-directional Recurrent Neural Network (Bi-RNN) with Long Short-Term Memory (LSTM), and LDA to capture contextual information and discover latent semantic information in the representation of short documents [Gu et al., 2018]. Finally, Pavlinek and Podgorelec suggested the use of a self-training algorithm within LDA to enlarge a small amount of pre-classified training data and achieve a semi-supervised learning process (ST-LDA, [Pavlinek and Podgorelec, 2017]). To the best of our knowledge, this was the first approach that utilising self-training in topic modelling. It is worth noting that the self-training process, although done automatically, can also be very time-consuming in LDA. As we shall see, our proposed topic modelling approach OLDA that incorporating an ontological component can significantly reduce the required training time whilst achieving higher classification accuracy at the same time.

3.2 Methodology

In this section, we describe our topic modelling approach addresses the two issues with LDA mentioned above, namely its inability to consider actual word semantical meanings, and the amount of human supervision needed to train the model. In order to address the first problem, we include an intermediate step to the LDA topic-modelling process using

$$\begin{pmatrix} d_{11}, d_{12}, \ldots, d_{1k} \\ d_{21}, d_{22}, \ldots, d_{2k} \\ \vdots \\ d_{n1}, d_{n2}, \ldots, d_{nk} \end{pmatrix} = \begin{pmatrix} q_{11}, q_{12}, \ldots, q_{1l} \\ q_{21}, q_{22}, \ldots, q_{2l} \\ \vdots \\ q_{n1}, q_{n2}, \ldots, q_{nl} \end{pmatrix} \times \begin{pmatrix} r_{11}, r_{12}, \ldots, r_{1m} \\ r_{21}, r_{22}, \ldots, r_{2m} \\ \vdots \\ r_{l1}, r_{l2}, \ldots, r_{lm} \end{pmatrix} \times \begin{pmatrix} s_{11}, s_{12}, \ldots, s_{1k} \\ s_{21}, s_{22}, \ldots, s_{2k} \\ \vdots \\ s_{m1}, s_{m2}, \ldots, s_{mk} \end{pmatrix}$$

Δ (documents/words) $\quad \Theta$ (documents/topics) $\quad \Sigma$ (topics/concepts) $\quad \Gamma$ (concepts/words)

Each document is a collection of words in Δ. Each concept is associated with corresponding words in Γ. Each topic is a distribution of concepts in Σ. The final topic model is Θ, where each document is a distribution of topics.

Fig. 3.2 *Ontology-Driven topic model matrices schematic*

concepts representing knowledge of ontology to capture the different meanings of the words. For this reason, our technique can be considered an *ontology-driven* variant of LDA, which we abbreviate to OLDA. In its simplest approach, it also requires some level of human supervision. But as for LDA, it also allows the incorporation of a self-training phase so that the number of human efforts can be reduced. We refer to this self-training variant as ST-OLDA.

Let $\mathscr{D} = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$ be a collection of documents to be classified into the topics $\mathscr{T} = \{T_1, \ldots, T_l\}$ and $\mathscr{W} = \{W_1, \ldots, W_j, \ldots, W_k\}$ is a set of words occurring in $\mathscr{D}$.

The overall matrix schema of OLDA is shown in Figure 3.2. OLDA's aim is to obtain a topic model, which is a documents/topics matrix Θ in Figure 3.2 giving the probability q_{xy} of each document D_x being about a certain topic T_y. The incorporation of the ontology concepts is done through the introduction of an intermediate concept dimension in the matrices as follows. We first pre-process the documents employing standard open-source NLP tools (StanfordNLP [Manning et al., 2014a]) for part-of-speech (POS) tagging and extracting the set $\mathscr{W}$ of all words in them. In the matrix Δ, each document is a collection of words. As before, we construct the matrix Δ of binary values, where each cell d_{ij} is given the value 1 if the document D_i contains the word $W_j \in \mathscr{W}$ or 0, otherwise. Using ConceptNet [Speer et al., 2017] or DBpedia [Auer et al., 2007], we then construct the set of all concepts $\mathscr{C}$ that are associated with a word $W \in \mathscr{W}$. Analogously, we then construct the matrix Γ of binary values, where each cell s_{ro} is given value 1 if the word $W_o \in \mathscr{W}$ can be described by the concept $C_r \in \mathscr{L}$, or 0 otherwise (this process is described in more detail in Section 3.2.1). In the matrix Γ, each concept is associated with corresponding words. The matrix Σ giving the probabilities r_{ab} of each topic T_a being described by each concept c_b is constructed using a *logistic regression technique*. When constructing Σ, each topic is represented by a distribution of concepts. Finally, Θ is computed by Δ, Σ and Γ following the schema (the computation of Σ and Θ are described in Section 3.2.2). In the result topic model Θ, each

document is represented by a distribution of topics. This topic model can then be used to train any classifier (C-SVM in our work) to perform *text classification*. In Section 3.3 we explain how the amount of human supervision can be minimised.

3.2.1 Generating the Concepts/Words Matrix Γ

ConceptNet is an open-source semantic network based on the knowledge from the Open Mind Common Sense (OMCS) database. ConceptNet is shown as a directed graph, where nodes as concepts, and edges as assertions of common sense about these concepts [Havasi et al., 2007]. Concepts represent sets of closely related natural language phrases, which could be noun phrases, verb phrases, adjective phrases, or clauses. In this work, we use two types of relationship between concepts: *IsA* and *RelateTo* to create the set of concepts $\mathscr{C}$ and the matrix Γ which gives the association between words and concepts as follows (this process is done programmatically via scripts without human intervention).

DBpedia is a crowd-sourced community website providing structured content extracted from the information created in various Wikipedia projects. This structured knowledge is freely available for use and described by a shallow, cross-domain ontology called the *DBpedia Ontology*. The DBpedia Ontology currently consists of 685 concepts described by 2795 different properties. An important property of each concept is its *Type*, which loosely describes the semantic meaning of the concept. Like ConceptNet, we use the type property of the concepts to create the set of concepts $\mathscr{C}$ and the matrix Γ.

Information Extraction from Web

The process of ontological concepts extraction from the web, such as ConceptNet and DBpedia, is called Internet information extraction. Instead of extracting concepts from APIs, we introduced a semi-supervised approach to obtain information directly from the web pages of ConceptNet and DBpedia via query answering, so that we could extract as much available information as possible.This is an important programming technique in computer science field aiming to extract records from web pages and identify items written in Hypertext Transfer Protocol (HTML), Javascript or PHP [Chu et al., 2015]. This technique can facilitate various applications such as data analysis and data integration and has been applied in different research areas such as semantic web [Kayed and Chang, 2010].

One of the key components in Internet information extraction is called the *wrapper*, which can be achieved by three approaches [Agarwal and Liu, 2008]: 1) *manual approach* requires a human observing the web page and the source code and manually find the data extraction patterns; 2) *wrapper induction* is a semi-automatic approach. It requires some

level of manual labelling, and then generating training templates and extraction rules through machine learning techniques; 3) *automatic extraction* is an unsupervised approach without any manual labelling involved, which can be scaled up to cover more websites for data extraction.

These three types of wrapper all can be used to extract two main types of data [Agarwal and Liu, 2008, Su et al., 2012]: 1) a group of data that is described in a similar format, HTML tags and rendered in a contiguous region, 2) a list of data presented in sub-trees and under the same parent node with similar repetition of HTML tag structures. Many frameworks designed for capturing these two types of data are published as research literature. Chang and Lui designed an Information Extraction based on Pattern Discovery (IEPAD) that can discover repetitive patterns by matching HTML tag strings and creating a tree structure of the extracted data [Chang and Lui, 2001]. In order to reduce computation time and improve the accuracy of the extracted data, Zhai and Liu designed a Data Extraction based on Partial Tree Alignment (DEPTA) that can consider not only the source code but also visual information with an alignment technique [Zhai and Liu, 2006].

We utilised an open-source automatic Internet information extraction system proposed by [Chu et al., 2015]. Their approach is trained in a semi-supervised manner to extract a list of data presented in sub-trees under key properties, such as *IsA, RelatedTo or Type* in our case. The data extraction system contains six main steps:

1. By querying each noun $W \in \mathscr{W}$ in ConceptNet or DBpedia web page server, each corresponding HTML document is syntactically parsed and converted into text strings. This can be done based on HTML tags or the stop sign of the natural languages such as a period.

2. Transforming a web page (HTML) into a Document Object Model (DOM) tree structure, which is a cross-platform and a language-independent convention. The system identifies the HTML tags and their function within the text strings and converts the HTML texts into a DOM. Each sub-tree in the DOM contains the texts within the start and end tags of the HTML.

3. Data Path Matching (DPM) is used to identify structural similarities of the sub-trees within the DOM structure. The conventional complex tree distance measure algorithms are used to discover similarities of the "Data Path". The key properties such as *IsA, RelatedTo or Type* are recorded.

4. Sub-trees that have fewer similarities are deleted, and sub-trees that are exactly the same (content overlapping) are merged together. Similar sub-trees are grouped into the

same group, and the group that is with the largest number of key properties is identified as the important sub-trees.

5. Path-Code-Sting Alignment technique is used to align corresponding data items, which transform records into HTML tag strings and measure their minimum distance between characters.

6. All the contents of the important sub-tree for each document associated with W are extracted from DOM, and they are the set of concepts $C(W)$.

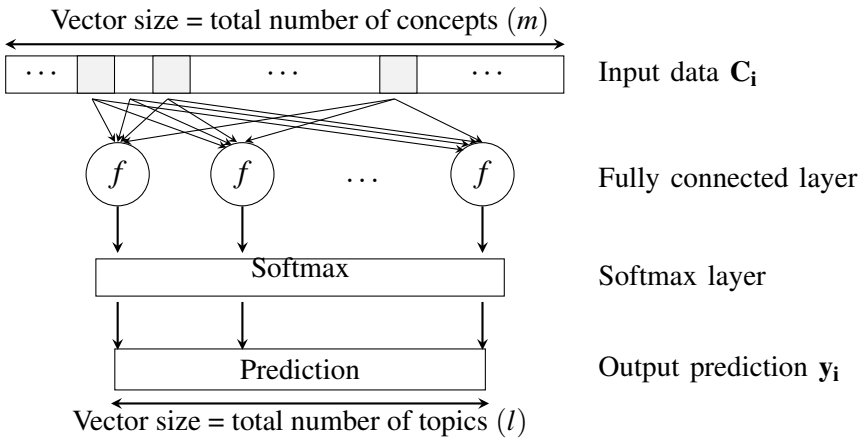
Generating the Concepts/Words Matrix Γ using extracted concepts

After extracting concepts of each noun $W \in \mathscr{W}$ from the ontological website and constructing the set of concepts $C(W)$ associated with the word W, we are ready to construct the Concepts/Words Matrix Γ. Because of the way the type properties are presented in the ontological website, we perform some basic data cleansing: we remove any redundant information in the property, segment words as needed, and aggregate similar terms. For example, the word "computer" has ten different Type properties in DBpedia: *Thing; Device; Artifact100021939; ComputerSystem103085915; Instrumentality103575240; Object100002684; PhysicalEntity100001930; System104377057; Whole100003553 and WikicatComputerSystems*. During cleansing, we remove the reference numbers from the types, segment words in terms such "ComputerSystem", and combine similar words such as "System" and "Systems" into a single concept. In this example, we would associate the word "computer" with the set of concepts $C(computer) = \{Thing, Device, Artifact, Computer\ System, Instrumentality, Object, Physical\ Entity, System, \ldots\}$.

We then set $\mathscr{C} = \bigcup_{W \in \mathscr{W}} L(W)$; assume a fixed ordering of concepts $[C_0, C_1, \ldots, C_m]$ (for $C_i \in \mathscr{C}$); and then construct the matrix Γ by setting $s_{ij} = 1$ if $C_j \in C(W_i)$, or 0 otherwise.

3.2.2 Generating the Matrices Θ and Σ

The documents/topics matrix Θ, and the topics/concepts matrix Σ are generated iteratively using the input matrix Γ in a logistic regression model. The model uses the linear weighted combination of inputs from Γ and generates the predicted probabilities of each concept relating to each topic (i.e., the matrix Σ) [Walker and Duncan, 1967a, Menard, 2002]. A schematic diagram of the model is shown in Fig. 3.3. The input of the logistic regression model is a vector of concepts $\mathbf{C_i}$ for word W_i. For each word $W_i \in \mathscr{W}$, the corresponding column in the concepts/words matrix Γ is used as an input data vector $\mathbf{C_i}$ (equation (3.4) below). The output of the logistic regression model is a prediction vector $\mathbf{y_i}$, which represents



The input is a vector of concepts $\mathbf{C_i}$ for word W_i. It is fed into a fully connected layer defined by Equation 3.5. The output of the fully connected layer is normalised by a softmax layer defined by Equation 3.6 to generate the final output $\mathbf{y_i}$, which is the predicted probability of each concept being associated with a topic

Fig. 3.3 *Structure of the logistic regression model*

the predicted probability of each concept being associated with a topic, as described next. A fully connected layer takes the vector $\mathbf{C_i}$ and generates the evidence vector $\mathbf{z_i}$ using (3.5) and a weight matrix $\mathbf{W}_t$ and bias vector b_t. The initial values $\mathbf{W}_0$ and b_0 are randomly given.

$$\mathbf{C_i}_j = \begin{cases} 1 & \text{if } c_j \in L(W_i) \\ 0 & \text{if } c_j \notin L(W_i) \end{cases} \tag{3.4}$$

$$\mathbf{z}_i = f(\mathbf{C_i}) = \mathbf{W}_t\mathbf{C_i} + b_t \tag{3.5}$$

Each element $\hat{r}_a$ in the evidence vector $\mathbf{z_i}$ is then normalised in the softmax layer to finally generate the vector $\mathbf{y_i}$ according to (3.6) (this means that the values within $\mathbf{y_i}$ add up to 1). Each element r_{ab} in the output vector $\mathbf{y_i}$ is the predicted probability of each concept c_b being associated with a topic T_a. This whole process is repeated for all words ($i \in (1,2,\ldots,k)$), resulting in the matrix Σ_t. Finally, the matrix Θ_t can be computed using the matrix schematic shown in Fig. 3.2.

$$\mathbf{y_i} = softmax(\mathbf{z}_i) = \frac{e^{\hat{r}_a}}{\sum_{a=1}^{l} e^{\hat{r}_a}} \tag{3.6}$$

Consequently, the initial matrices Σ_0 and Θ_0 are obtained using random values for the weight matrix $\mathbf{W}_0$ and the bias vector b_0. For each subsequent iteration $t+1$, we then measure the Euclidean distance between the predicted classification Θ_t and the true classification Θ_s

(recall Θ_s is manually done). Using the Stochastic Gradient Descent technique we obtain new values for $\mathbf{W}_{t+1}$ and the vector b_{t+1} [Kiefer et al., 1952, Bottou, 1998] that minimise this distance. We then calculate Σ_{t+1} and Θ_{t+1} as before using $\mathbf{W}_{t+1}$ and b_{t+1}. This process continues until the distance between the predicated classification Θ_j computed in an iteration j and the true classification Θ_s goes below a desired threshold. The output of this process is the documents/topics matrix Θ, the topics/concepts matrix Σ, and the optimised weight matrix $\mathbf{W}$ and bias vector b.

3.3 Topic Classification with Self-Training

Obviously, obtaining a large training dataset is costly, so we would like to minimise the amount of pre-classified data required. As done for LDA in the creation of ST-LDA, this can be done by introducing a self-training stage to enlarge the original amount of manually trained data.

In this section, we consider the use of two self-training approaches for this: the first, presented in Section 3.3.1, consists of a relatively ad hoc procedure that is quick to perform and produces good but sub-optimal results. As an alternative, we also describe Pavlinek et al.'s self-training procedure [Pavlinek and Podgorelec, 2017] in Section 3.3.2. We will see that this procedure takes nearly twice as long to complete as the ad hoc method, but produces the best results when combined with OLDA. We stress that the introduction of the concept matrix reduces the time required for training by approximately half independently of the training procedure used.

3.3.1 A Simple Self-training Procedure

Instead of using a large amount of pre-classified documents to calculate values for the weight matrix $\mathbf{W}$ and the bias vector b, the idea is to take a very small amount of pre-classified documents D_s and a much larger amount of unclassified documents D_u to output an enlarged classified training matrix Θ_{ss} from the manually provided matrix Θ_s. $\mathbf{W}$ and b are obtained from D_s and Θ_s as described in Section 3.2.2 until we reduce the distance between Θ_t and Θ_s to below a pre-defined distance threshold DT. Here, the distance is calculated as Euclidean distance, as shown in Equation 3.7.

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.7}$$

Data: D_s, D_u, DT
Result: D_{ss}
while $d_{euc} > DT$ **do**
  foreach $d_i \in D_s$ **do**
    Perform Logistic Regression model.
    Measure Euclidean distances d_{euc} between Θ_t and Θ_s.
    Adjust **W** and b.
  end
  Move selected documents $d_j \in D_u$ to D_s based on obtained **W** and b.
end

Algorithm 1: Algorithm for the ad hoc self-training

In a second phase, we use the values of **W** and b thus obtained to automatically train the remaining unclassified data (D_u). Thus, the final training set D_{ss} consists of the manually classified set D_s together with the automatically trained set D_u and is applicable for training purposes as in any other supervised classification method. The resulting topic model Θ_{ss} and Σ_{ss} can then be used to classify the remaining unclassified documents.

3.3.2 Advanced Self-training Procedure

Pavlinek et al. proposed a more elaborate self-training algorithm also consisting of two phases [Pavlinek and Podgorelec, 2017]. As before, the goal of the first phase is to generate a topic model from the smaller amount of manually classified data D_s. However, they employ Gibbs sampling [Casella and George, 1992] to do this.

In the second phase, unclassified data (from D_u) is iteratively classified using the topic model generated in the first phase and compared using a centroids distance until a predefined threshold is reached. The centroid distance is defined by a semantic similarity measure based on the topic distribution and a cosine similarity measure defined in terms of the centroids for each topic category [Han and Karypis, 2000]. Each iteration is performed in two steps:

- For each topic category T_a, a centroid vector CV is created in terms of the average of pre-classified documents $d_i(T_a) \in D_s$ in the given category T_a. Then the cosine distance between unclassified data $d_j \in D_u$ and centroid vectors $CV(T_a), (1 \leq a \leq l)$ are computed. Here, the cosine distance is defined as a complement measure to cosine similarity as shown in Equation 3.8.

$$d_{cos}(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3.8}$$

Data: D_s, D_u, ST
Result: D_{ss}
while $\varepsilon > ST$ **do**
  foreach $T_a \in \mathscr{T}, 1 \leq a \leq l$ **do**
    Create centroid vectors $CV = cv_1, cv_2, \ldots, cv_l$ based on pre-classified documents D_s.
  end
  foreach $d_j \in D_u$ **do**
    Measure cosine distances $d_{cos}(d_j, cv_i), cv_i \in CV$.
    Calculate the difference between two minimum cosine distances.
    $\overline{CV} = \{cv_x \in CV | \exists cv_y \in CV : d_{cos}(d_j, cv_x) \leq d_{cos}(d_j, cv_y)\}, CV \subset \overline{CV}$
    $cv_{min_1} = argmin_{cv_i \in CV}(d_{cos}(d_j, cv_i))$
    $cv_{min_2} = argmin_{cv_i \in \overline{CV}}(d_{cos}(d_j, cv_i))$
    $dif_j = d_{cos}(d_j, cv_{min_2}) - d_{cos}(d_j, cv_{min_1})$
    Sort unclassified documents based on the differences from the highest to the lowest.
  end
  Define $\varepsilon = max(\{dif_1, \ldots, dif_n\})$
  if $\varepsilon > ST$ **then**
    Move selected documents $d_j \in D_u$ to D_s, where $T(d_j) = T_{cv_{min_1}}$
  end
end

Algorithm 2: Algorithm for Pavlinek self-training

- Unclassified documents in D_u are then sorted by the difference between distances from the two nearest centroids. The higher rank means the document is much closer to the nearest centroid than to the next one. The unclassified document with the highest rank, i.e. the most reliable document, are classified as the topic category according to its nearest centroid.

The second phase finishes when for each unclassified document in D_u, the difference between the distances from the two nearest centroids is smaller than the similarity threshold ST. As a result of this phase, we also end up with an enlarged classified training set D_{ss} consisting of the manually classified set D_s and the automatically classified set D_u. Full details of the whole process are shown in Algorithm 2.

3.4 Experimental Analysis

As we mentioned, OLDA can be used with or without a self-training stage. When self-training is employed, we use the prefix "ST" and refer to the resulting classification method as ST-OLDA instead. As we suggested two different self-training procedures, the prefix ST

ifs subscripted with H (to indicate the use of the ad hoc training procedure) or P (to indicate the use of Pavlinek et. al.'s). Where the distinction is irrelevant, we avoid the subscript. With all this in mind, we conducted comprehensive benchmarking to evaluate the performance of our method and variants against a number of other semi-supervised methods using four different widely available datasets.

More specifically, we compared the performance of OLDA with the performance of the Expectation-Maximisation and Naïve Bayes classifier (EM-NB) [Nigam et al., 2000]. We also compared the performance of ST-OLDA with the Bag-of-Words (BOW) representation with a Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme [Robertson, 2004], with the Latent Dirichlet Allocation (LDA) [Pavlinek and Podgorelec, 2017] and with OntoLDA [Allahyari and Kochut, 2015]. Furthermore, we considered the two self-training techniques (ST_H and ST_P) described in Section 3.3 for OntoLDA, LDA, TF-IDF and OLDA, resulting in a total of six variations of the semi-supervised methods. In addition, we compared our ST-OLDA with state-of-the-art word embedding based approaches [Fu et al., 2016, Liu et al., 2019] and knowledge-based approach [Allahyari and Kochut, 2015].

3.4.1 Experimental Setup

To perform a comparative analysis, four fully classified training datasets (presented below) were used. Each dataset was split into a training dataset (50%-70%) and a testing dataset (30%-50%). For each round of supervised experiments, all the training datasets were used to construct the topic model and the supervised classification method SVM. For each round of the semi-supervised experiments, 10% of the training datasets denotes the initial pre-classified dataset D_s, and the remaining training data form the unclassified datasets D_u. The self-training topic model was then used to prepare the final classified datasets D_{ss}, which were trained with a supervised classification method SVM. The trained classifiers were finally evaluated on the testing datasets.

Both self-training algorithms were implemented in Java using WEKA [Hall et al., 2009], which is an open-source machine learning environment. LibSVM implementation was used to train an SVM classifier with a linear kernel on the final classified dataset [Chang and Lin, 2011]. The LDA topic models were constructed using the MALLET toolkit [McCallum, 2002]. All experiments were performed on a PC with an i7 processor, an NVIDIA GeForce GPU GTX 970M graphics card, and 16GB RAM.

3.4.2 Datasets Used in the Analysis

In our analysis we used the 20 Newsgroups dataset [2], the Reuters R8 and R52 datasets [3] and the WebKB dataset [4]. For each dataset, we performed some pre-processing to combine word variants and to remove words that we deemed irrelevant. To be precise: (i) all words were converted to lower case; (ii) stop words (such as "etc.", "I'm" and "of") were removed; (iii) words shorter than three characters were also removed; and (iv) using lemmatisation tool from StanfordNLP, for example, plural words were converted into singular. We now briefly describe how each of these datasets was used.

20 Newsgroups

This dataset comprises a collection of $18,846$ newsgroup documents, partitioned (nearly) evenly across 20 different categories, each corresponding to a different topic. We used the so-called "bydate" version, where duplicates and some headers are removed. In total, $233,745$ words were extracted from the documents. After pre-processing the number of words was reduced to $155,387$ and hence our documents/words matrix Δ is a $18,846 \times 155,387$ binary matrix. We then extracted concepts from ConceptNet and DBpedia. $13,591$ concepts associated with these words from ConceptNet were extracted, yielding a concepts/words matrix Γ of size $13,591 \times 155,387$ for the ST-OLDA methods. $13,820$ concepts associated with these words from DBpedia were extracted, yielding a concepts/words matrix Γ of size $13,820 \times 155,387$ for the ST-OLDA methods. To have a common baseline for comparison, we randomly selected 50% of the training data for EM-NB and OLDA, leaving the remaining 50% for testing. For each round of experiments for ST TF-IDF, ST-LDA, ST-OntoLDA and ST-OLDA, we used 5% of the data for the first phase of training, 45% for the second semi-supervised phase, and used the remaining 50% for testing.

Reuters R8

Reuters R8 is derived from the Reuters-21578 dataset and is singly labelled with a ModApte split, which means that each topic category contains at least two documents and hence at least one can be used for training and one for testing. Reuters R8 contains 7674 documents divided into 8 categories. From the words extracted, 7808 were left after pre-processing, resulting in a 7674×7808 binary documents/words matrix Δ. We then found 5289 concepts associated with these words in ConceptNet, yielding a 5289×7808 concepts/words matrix

[2] http://qwone.com/ jason/20Newsgroups/
[3] https://www.cs.umb.edu/ smimarog/textmining/datasets/
[4] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

Γ. As for DBpedia, 5315 concepts associated with these words were extracted, yielding a 5315×7808 concepts/words matrix Γ. With this dataset, we employed an approximate 70/30 ratio for training/testing as normally employed elsewhere. For each round of supervised experiments, we randomly selected 70% of the data for the training phase of EM-NB and OLDA, leaving the remaining 30% for testing. For ST TF-IDF, ST-LDA, ST-OntoLDA and ST-OLDA, we used 7% of the data for the first phase of training, 63% for the second phase, leaving the remaining 30% for testing.

Reuters R52

As for Reuters R8, Reuters R52 is also derived from the Reuters-21578 dataset, whereas Reuters R52 consists of 9100 documents divided into 52 categories. Pre-processing of the words extracted resulted in 8937 words yielding a 9100×8937 binary documents/words matrix Δ. As before, we extracted 6291 associated concepts from ConceptNet, yielding a 6291×8937 concepts/words matrix Γ. As for DBpedia, we extracted 6471 associated concepts, yielding a 6471×8937 concepts/words matrix Γ. The partition of training sets and testing sets are the same as Reuters R8 shown above.

WebKB

This dataset comprises a collection of websites from computer science departments, whose pages are divided into seven categories: student, faculty, staff, course, project, department and other. Our experiments used a variant of the dataset covering 4199 documents from the first four previous categories. After pre-processing, we were left with a 4199×7719 binary documents/words matrix Δ. We found 5099 associated concepts in ConceptNet, yielding a 5099×7719 concepts/words matrix Γ. We found 5109 associated concepts in DBpedia, yielding a 5109×7719 concepts/words matrix Γ. For each round of experiments, we randomly selected approximately 66% of the training data for EM-NB and OLDA, leaving the remaining 34% for testing. For ST TF-IDF, ST-LDA, ST-OntoLDA and ST-OLDA, we used 6% of the data for the first phase of training, 60% for the second phase, and the remaining 33% for testing.

3.4.3 Experimental Results

For concepts extracted from ConceptNet and DBpedia, we both conducted two rounds of experiments with each of the four datasets. For the supervised approaches, we skipped the self-training phase and used the proportions of data described in Section 3.4.2. In each round of the semi-supervised experiments, we performed 10 repetitions in training and selected the

Table 3.1 Classification accuracy results (Confidence Interval (CI)=95%)

Dataset			20Newsgroup	Reuters R8	Reuters R52	WebKB
Supervised	EM-NB		53.12%	34.12%	26.90%	55.56%
	OLDA	ConceptNet	**89.86%**	**87.47%**	**67.11%**	**85.21%**
		DBpedia	**90.22%**	**87.90%**	**68.02%**	**85.87%**
Semi-supervised ST_H	TF-IDF		56.21%	20.11%	22.05%	63.31%
	LDA		61.33%	60.54%	45.88%	68.09%
	OntoLDA	ConceptNet	66.76%	68.79%	49.12%	70.68%
		DBpedia	67.15%	69.06%	49.81%	71.33%
	OLDA	ConceptNet	**71.76%**	**75.89%**	**56.02%**	**74.75%**
		DBpedia	**72.12%**	**77.32%**	**56.14%**	**74.90%**
Semi-supervised ST_P	TF-IDF		60.25%	23.66%	25.87%	67.13%
	LDA		68.51%	75.71%	53.24%	72.38%
	OntoLDA	ConceptNet	72.07%	79.93%	58.98%	74.60%
		DBpedia	72.41%	80.20%	59.75%	75.53%
	OLDA	ConceptNet	**77.67%**	**82.86%**	**64.08%**	**80.79%**
		DBpedia	**78.01%**	**83.11%**	**64.25%**	**81.89%**

Table 3.2 Time to construct the 20 Newsgroups topic model

Technique		Construction time (days)
Supervised	EM-NB	30
	OLDA	8
Semi-supervised ST_H	TF-IDF	2
	LDA	5
	OntoLDA	2
	OLDA	2
Semi-supervised ST_P	TF-IDF	6
	LDA	10
	OntoLDA	5
	OLDA	5

data for training using stratified random sampling for each topic category, so that each topic had equal representation in the training set.

We compared the proposed OLDA and ST OLDA topic model with the following five methods and variants:

- EM-NB: For the supervised experiments, we conducted an EM-NB model based on the work of Nigam et al. [Nigam et al., 2000].
- ST TF-IDF: Next, we conducted a TF-IDF model by representing each document with a bag of words and TF-IDF weighting following the work of Sriurai [Sriurai, 2011]. For further investigate, we also combined this model with a self-training procedure, denoted as ST TF-IDF. The similarity threshold ST was set as 0.1.
- ST LDA: We conducted an LDA model following the work of Pavlinek and Podgorelec [Pavlinek and Podgorelec, 2017]. Standard LDA topic model from MALLET toolkit was used. The number of Gibbs iterations was set to 500. As for their work, we combined this model with a self-training procedure, denoted as ST LDA. The similarity threshold ST was set as 0.1.
- ST OntoLDA: Finally, we conducted an OntoLDA topic model following the work of Allahyari and Kochut [Allahyari and Kochut, 2015]. The number of Gibbs iterations was set to 500. For further investigate, we also combine this model with a self-training procedure, denoted as ST OntoLDA. We also employed concepts from ConceptNet and DBpedia to evaluate the performances of different ontologies.

The supervised OLDA, ST TF-IDF, ST LDA, ST OntoLDA and ST OLDA topic models were evaluated with C-SVM classifiers. Table 3.1 summarises the classification accuracy

results of EM-NB, TF-IDF, LDA, OntoLDA and OLDA when using supervised training procedure and either of the two self-training procedures ST_H and ST_P. Table 3.2 summarises the topic model's construction times for each technique for the 20Newsgroup dataset.

In what follows, we discuss the results using each self-training procedure in more detail.

Supervised Approaches

As we mentioned, we can skip the self-training phase in our method resulting in a fully supervised classification engine that we simply refer to as OLDA (our baseline). We compared OLDA's accuracy with that of the supervised EM-NB approach [Nigam et al., 2000]. For concepts extracted from ConceptNet and DBpedia, our results show that OLDA outperformed EM-NB in all datasets. As shown in Table 3.1, OLDA outperforming EM-NB by quite a considerable margin (e.g., with ConceptNet ontology, 26.90% against 67.11% in the Reuters R52 dataset; with DBpedia ontology 68.02% in the same dataset). With either ConceptNet or DBpedia ontology, OLDA achieves around the same accuracy in all four datasets. With around the same amount of concepts from either ontology included in OLDA, a similar classification accuracy results can be achieved. This shows that OLDA can be easily applied in different domains with different ontologies.

As shown in Table 3.2, the construction of the topic model for the 20 Newsgroups dataset using OLDA only took about 8 days to complete while it took 30 days for EM-NB. This shows the efficiency of the OLDA.

Self-Training Using the Simplified Approach (ST_H)

As we mentioned in Section 3.3.1, the training procedure stops when the distance between the predicted and actual classification drops below a certain threshold. In our experiments, this distance drops dramatically in the first $2,500,000$ iterations, decreasing further but at a reduced rate in later iterations. The distance remained fairly stable after $20,000,000$ iterations dropping to values close to 0.54. For that reason, we stop iterating when the distance goes below 0.55. As for LDA and OntoLDA, Gibbs sampling algorithm was run for 500 iterations.

As shown in Table 3.1, TF-IDF performed worst of all in all datasets, and OLDA also outperforming LDA by quite a considerable margin (e.g., with ConceptNet, 75.89% against 60.54% in the Reuters R8 dataset; with DBpedia ontology, 77.32% against 60.54% in the same dataset). This shows the advantage of introducing the ontology intermediate matrix. With either ontology, OLDA outperforms OntoLDA, which confirms the improvements of Logistic Regression model (e.g., with ConceptNet, 75.89% against 68.79% in the Reuters R8

Table 3.3 Topic classification results of state-of-the-art work on 20Newsgroup dataset

Model		Accuracy
Word embedding based	LFLDA [Fu et al., 2016]	80.94%
	<word, POS> embedding model [Liu et al., 2019]	83.05%
Knowledge based	ST_P-OntoLDA [Allahyari and Kochut, 2015]	72.41%
	ST_P-OLDA	78.01%

dataset; with DBpedia ontology, 77.32% against 69.06% in the same dataset). In addition, OLDA achieves around the same accuracy with both ontologies. This shows that OLDA can be applied in different domains with different ontologies. As shown in Table 3.2, the construction of the topic model for the 20 Newsgroups dataset using the training procedure ST_H for OLDA and OntoLDA took about two days to complete while it took five days for LDA. The introduction of the ontology reduces the training procedure of topic model by 40%.

Self-Training Using Pavlinek et al.'s Approach (ST_P)

OLDA's and OntoLDA's construction of the topic model for the 20Newsgroup dataset using the training procedure ST_P took about five days, whilst LDA's took ten days. That is, OLDA's construction took around half the time because of the ontology intermediate matrix.

In terms of accuracy, the training procedure ST_P performed better in all techniques and datasets. TF-IDF performed worst in all datasets, albeit it was better when using the training procedure ST_P than when using ST_H. The best combination was ST_P and OLDA, which outperformed ST_P and LDA by quite a considerable margin (e.g., with ConceptNet, 64.08% against 53.24% in the Reuters R52 dataset; with DBpedia, 64.25% against 53.24% in the same dataset).

So we can conclude that the self-training procedure ST_P is superior to the simple training procedure ST_H although its topic model takes roughly twice as long to construct. We can also conclude that the introduction of the intermediate ontology concepts to the topic model helps to reduce the amount of time required to train the model (independently of the self-training procedure employed).

Table 3.3 compares the classification accuracy results of ST_P-OLDA against some state-of-the-art work on 20Newsgroups dataset: including both word embedding based approaches (LFLDA [Fu et al., 2016] and <word, POS> embedding model [Liu et al., 2019]) and knowledge-based approaches (ST_P-OntoLDA [Allahyari and Kochut, 2015]). Even tough ST_P-OLDA achieves slightly lower accuracy compared to LFLDA and <word, POS> embedding

model, our proposed ST_P-OLDA still benefits when dealing with small corpus. The inclusion of ontology component is able to capture more semantical meanings of words regardless of the context. As for word embedding based approaches depending on external word embedding tools, it is difficult for them to deal with words that are not included in the external word embeddings. Comparing against the state-of-the-art knowledge-based approach ST_P-OntoLDA, the introduction of the concept matrix into the topic model not only increases the accuracy of the classification across all datasets but also helps to reduce the training time by up to 60%.

3.5 Summary

Conventional data-driven approaches to topic modelling of natural language texts, such as Term Frequency - Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA), come with two important limitations. Firstly, these approaches do not use the semantical meanings of the words, ignoring the fact that individual words may have multiple meanings and that different words may have the same meaning. This limits the ability of the method to perform the modelling independently of the particular set of words describing the topics. Secondly, they require a significant amount of classified training data for supervised machine learning. Generating this training data is expensive and time-consuming as it relies on humans to collect, read and manually classify the data in a consistent manner.

In this chapter we propose a novel approach based on LDA that uses ontological information obtained from DBpedia about the semantical meaning of the words, allowing topics to be represented more faithfully and independently to the particular set of words used to describe them. This approach, called Ontology-Driven Latent Dirichlet Allocation (OLDA), can be combined with a self-training phase to produce a semi-supervised method (ST-OLDA), which requires only a small amount of pre-classified training data. The idea is to generate the topic model using the restricted amount of manually classified data – typically, only 10% of the training data, and then use the remaining 90% of the training data to automatically train the model. The resulting model is then used to classify the remaining testing data.

Our experiments, using the four datasets “20 Newsgroups”, “Reuters R8”, “Reuters R52" and “WebKB”, show that the addition of the semantical component into LDA significantly increases the accuracy of the classification. When used with different ontologies, OLDA achieves around the same results in supervised training and semi-supervised training. In addition, when used with self-training, this allows the reduction of the amount of trained data

needed and significantly increases the performance of the classification over ST-OntoLDA, ST-LDA and ST TF-IDF, while reducing the time required for training.

Our main conclusions can be summarised as follows:

1) The inclusion of the ontological component reduces the self-training time by nearly half using two distinct self-training procedures. In particular, it reduces the time needed for training using the self-training procedure proposed by [Pavlinek and Podgorelec, 2017] by nearly half in the 20 Newsgroups dataset.

2) The inclusion of the ontological component also increases the accuracy of the classification regardless of the self-training method employed by between 6 and 17 percentual points (depending on the training method and dataset).

3) Ontologies from different resources achieves around the same results with OLDA in both supervised training and semi-supervised training. OLDA can be easily applied in different domains with different ontologies.

4) The self-training procedure proposed by [Pavlinek and Podgorelec, 2017] produces better accuracy results than an Ad Hoc procedure suggested in this chapter, for LDA, OntoLDA and OLDA, independently of the dataset, although it takes twice as long to train. When combined with OLDA it provides the best accuracy results in all datasets, significantly outperforming ST-LDA.

These results are very encouraging, and we think that there is scope for further improvement of the classification accuracy by incorporating the relationships between words and ontological concepts into the topic model. In order to achieve this, a relation extraction algorithm is developed and described in the next chapter.

Chapter 4

Multiple-Relation Extraction from Single Sentences

In this chapter, we describe a relation extraction algorithm to extract structured information from unstructured texts written in natural language. These relationships between words can enable us to capture the semantical structures of texts rather than solely the symbolical structures. They can then be included into the topic model as additional, richer features for generating more meaningful topics. This chapter contributes to ongoing efforts to develop mechanisms for automated knowledge extraction from textual data.

While this work benefits a broad range of potential applications, our ultimate goal for *relationship extraction* is the construction of networks from textual data representing various associations among entities. Before this is achievable, a number of challenges need to be overcome. For example, the sentence

> *"The quality of magnesium status directly influences the Biological Clock function (BC)."* [Durlach et al., 2005]

describes a relationship between the entities *magnesium* and *Biological Clock function*. In order to recognise and automatically extract this relationship from the sentence, one needs to perform several tasks. Firstly, the grammatical structure of the sentence needs to be analysed using Natural Language Processing (NLP) techniques. A number of freely available software libraries and toolkits can be used to perform this analysis. Two important ones are Stanford's CoreNLP [Manning et al., 2014a] [1] and Apache's OpenNLP [Kottmann et al., 2011] [2]. They provide a rich set of tools for common NLP tasks such as tokenisation, lemmatisation, part-of-speech (POS) tagging, parsing, etc.

[1] http://stanfordnlp.github.io/CoreNLP
[2] https://opennlp.apache.org

A second important challenge, which has recently seen tremendous progress, is *entity recognition*. Existing Named Entity Recognition (NER) tools can recognise not only general terms such as proper nouns but also more specific entities such as diseases and symptoms [Carpenter, 2007, Settles, 2005, Aronson, 2001, Subramaniam et al., 2003]. However, pronouns are frequently used to refer to a previously mentioned entity, and the existing NER tools cannot make the corresponding references, failing to extract some entities.

A third task is the *relationship extraction* itself. Techniques for relationship extraction can be categorised into four distinct approaches: (*i*) extraction based on *co-occurrence* extraction, (*ii*) extraction using *pattern-based* approaches, (*iii*) extraction using *machine learning* and (*iv*) *rule-based* extraction [Skusa et al., 2005, Zweigenbaum et al., 2007]. [3] The first three of these techniques can only deal with simple relations between two entities connected by a target word and generally achieve relatively low precision and recall. Applying them in different domains can be time-consuming. Rule-based extraction normally achieves higher precision and can be applied in a variety of domains [Sharma et al., 2010].

Until recently, rule-based approaches could only extract a single relation embedded in a sentence composed of a verb phrase between a pair of entities of interest. This approach works well when extracting simple co-occurrence relations such as *Entity-Verb-Entity*. However, if a sentence contains multiple relationships embedded in complex structures, such as clauses structure and conjunctive structures, existing conventional rule-based algorithms may fail to capture all the relationships. To address this problem, we proposed an algorithm that extended the capability of conventional rule-based algorithms in two significant ways [Hao et al., 2017]. Firstly, unlike conventional single-relation algorithms which can only identify target verb phrases using POS tagging and parsing, our enhanced algorithm was able to capture more relations by using synonyms of verbs (as obtained from WordNet [University, 2010] and VerbNet [Schuler, 2005]) **(CON1)**. Secondly, we tackled multiple relationship extraction by dealing with sentences in which these relationships were embedded within three special types of sentence structure **(CON2)**: (*i*) those in which the relations were connected by a relative pronoun such as *which* or *that*; (*ii*) relations embedded in sentences connected by conjunctions such as *and* and *but*; and (*iii*) one-to-many and many-to-many relations expressed within the phrase level conjunctive structure.

In addition to these two extensions, we further enhanced the algorithm proposed in [Hao et al., 2017] in three ways. Authors often use pronouns to refer to entities within a certain context to keep coherence and avoid tautology. For example, the sentence

[3] These techniques are discussed in more detail in Chapter 2.2.

> *"Magnesium is an essential micronutrient for the human body, and its deficiency has been associated with risk of noncommunicable diseases."* [Hermes Sales et al., 2014]

describes a relationship between entities *Magnesium deficiency* and *noncommunicable diseases* via the pronoun *"its"*. However, a traditional relationship extraction procedure will fail to extract the relationship because it does not associate the pronoun "its" with the entity *Magnesium deficiency*, which can be recognised by the NER tool. For the third extension, we introduce a co-reference resolution component to form chains between pronominal words and the corresponding nominal words. Our algorithm then replaces the pronouns with their corresponding nominal words to obtain pronoun-free sentences whose bio-entities can then be correctly recognised by the NER tool.

Relations often are embedded in noun-preposition phrases. For example, the sentence

> *"These results suggest a profound effect of the combined supply of Mg and Mn on the biosynthesis of terpenes and phenolics."* [Farzadfar et al., 2017]

describes a relationship between the entities *Mg and Mn* and *biosynthesis of terpenes and phenolics*. However, the entities are connected not by a verb but by the noun-preposition phrase *"a profound effect of the combined supply of ... on ..."*. To deal with these scenarios, we propose a fourth extension to recognise sentences with unconventional structures and extract relations connected by noun-preposition phrases.

Finally, many sentences only describe the work of the publication rather than a relationship when using noun-preposition phrases. For example, the sentence

> *"The objective of this study was to determine the effect between vitamin D status and broad gene expression in healthy adults."* [Hossein-Nezhad et al., 2013]

only describes the work itself rather than a relationship between "vitamin D status" and "gene expression". In sentences such as the one above, the use of certain verbs may interfere with the structure of relations, and some adjectives and adverbs may modify the intended meaning of the relationship being expressed. In some cases, they could even invert the meaning that is suggested by the verb alone. We propose a fifth extension for relationship extraction that looks for specific verbs, adjectives and adverbs modifying the structure of relationships to extract them with an associated *positive*, *negative* or *neutral* "polarity". In the example above, the polarity of the embedded relationship would be neutral, which means that it would be extracted, but not explicitly recorded. Similarly, sometimes a relationship may be expressed in the opposite way that the verb alone would suggest. For example, the sentence

> "*These results reduce the possibilities of competitive inhibitory interactions between the mutant and wild-type ChlI1a and ChlI1b proteins.*" [Campbell et al., 2015a]

expresses a negative relationship between the entities "mutant ChlI1a and ChlI1b" and "wild-type ChlI1a and ChlI1b", which would be extracted but again not recorded.

To summarise, the algorithm presented in this chapter offered five main contributions to existing single-relation extraction algorithms: *i)* it added the ability to extract relationships embedded in semantically similar verbs **(CON1)**; and *ii)* it added the ability to extract multiple relationships embedded within certain types of sentence structures **(CON2)**; *iii)* our algorithm replaces pronouns with their corresponding bio-entities allowing the extraction of relationships that would otherwise be missed **(CON3)**; *iv)* the algorithm can also extract relationships embedded within noun-preposition phrases **(CON4)**; and *v)* once these relationships are extracted, a further refinement allows us to determine the relationship polarity and fine-tune the process by excluding relationships that have not been explicitly asserted in the text **(CON5)**.

The algorithm with only **(CON1-2)** was published as a conference paper entitled *" A verb-based algorithm for multiple-relation extraction from single sentences"* in the proceedings of the 2017 International Conference on Information and Knowledge Engineering [Hao et al., 2017]. **(CON3)** makes the relationship extraction independent of the existence of the pattern *Entity-Verb-Entity*, relying solely on the existence of multiple entities within a sentence. Thus, we claim that the extraction algorithm employing this enhancement is *entity-based algorithm*.

In order to evaluate the effectiveness of the contributions, we used the conventional rule-based algorithm as a baseline and measured its performance against the algorithm enhanced with a full range of combinations of contributions **(CON1,2,3,4,5)** over two datasets, one from biomedical domain containing 600 sentences and one from general domain containing 3,232 sentences. The detailed set of measurements is given in Table 4.9. In summary, the conventional rule-based algorithm without any improvements achieved an overall precision of 0.724 and had 0.643 recall. The addition of contributions **(CON1,2)** improved the extraction results giving an overall precision of 0.884 and 0.817 recall. Finally, the entity-based algorithm with contributions **(CON1,2,3,4,5)** achieved an overall precision of 0.914 with 0.94 recall, therefore offering significant advantages over the conventional rule-based algorithm.

It is worth mentioning that the entity-based algorithm can be applied to different domains as long as the embedded NER and verb detection components utilise corpora appropriate for the domains. The entities are recognised by standard NER tools that can be trained for

corpora of different domains. Similarly, the verbs are identified from a fixed database. In this chapter, we used *Universal Medical Language System* (UMLS) for the biomedical domain, VerbNet for the general domain. WordNet is also used to expand the list of main verbs.

The rest of the chapter is organised as follows. Section 4.1 presents a general overview of our entity-based algorithm for relation extraction. Section 4.2 describes the data collection and pre-process procedure. Section 4.3 describes the Named Entity Recognition process and followed by relationship extraction process presented in Section 4.4. Section 4.5 explains the polarity adjustment procedure. Section 4.6 provides an evaluation of the algorithm itself and this is followed in Section 4.7 with a discussion on issues left as future work. We then summary in Section 4.8 with some final remarks.

4.1 Methodology Overview

As mentioned above, our algorithm extracts relationships from documents. Each text is divided into sentences, and each sentence is then processed by standard NLP techniques for POS tagging and parsing. Then, the pronominal reference of each sentence is analysed. Sentences with pronouns are processed by a pronominal replacement algorithm to obtain equivalent sentences without pronouns. NER is applied to identify the relevant entities for sentences without pronouns. A database of interest, such as NLPBA or BioCreative, is used to train the NER tool to recognise "target" biomedical terms such as "magnesium deficiency" or "migraine attack". The verbal structure of each sentence is also analysed and extracted. If a sentence is verb-centric, main verbs are extracted, which are semantically similar to one from the UMLS, WordNet and VerbNet lists. For example, the sentence

> *"Female hormones lower magnesium but increase calcium levels which enhance migraine ubiquitousness."* [Dhillon et al., 2011]

is a verb-centric sentence [Hao et al., 2017] containing three recognised main verbs *lower, increase and enhance*.

If the sentence contains noun-preposition phrases, relation connection words are extracted based on rules we created. For example, the sentence

> *"These results suggest a profound effect of the combined supply of Mg and Mn on the biosynthesis of terpenes and phenolics."* [Farzadfar et al., 2017]

contains a relationship between *Mg and Mn* and *biosynthesis of terpenes and phenolics* connected by a noun-preposition phrase *"a profound effect of ... on ..."*.

Once the algorithm constructs the tuples *Entity| Relation connection | Entity*, it checks for any modifiers that might affect the polarity of the resulting extracted relation based on

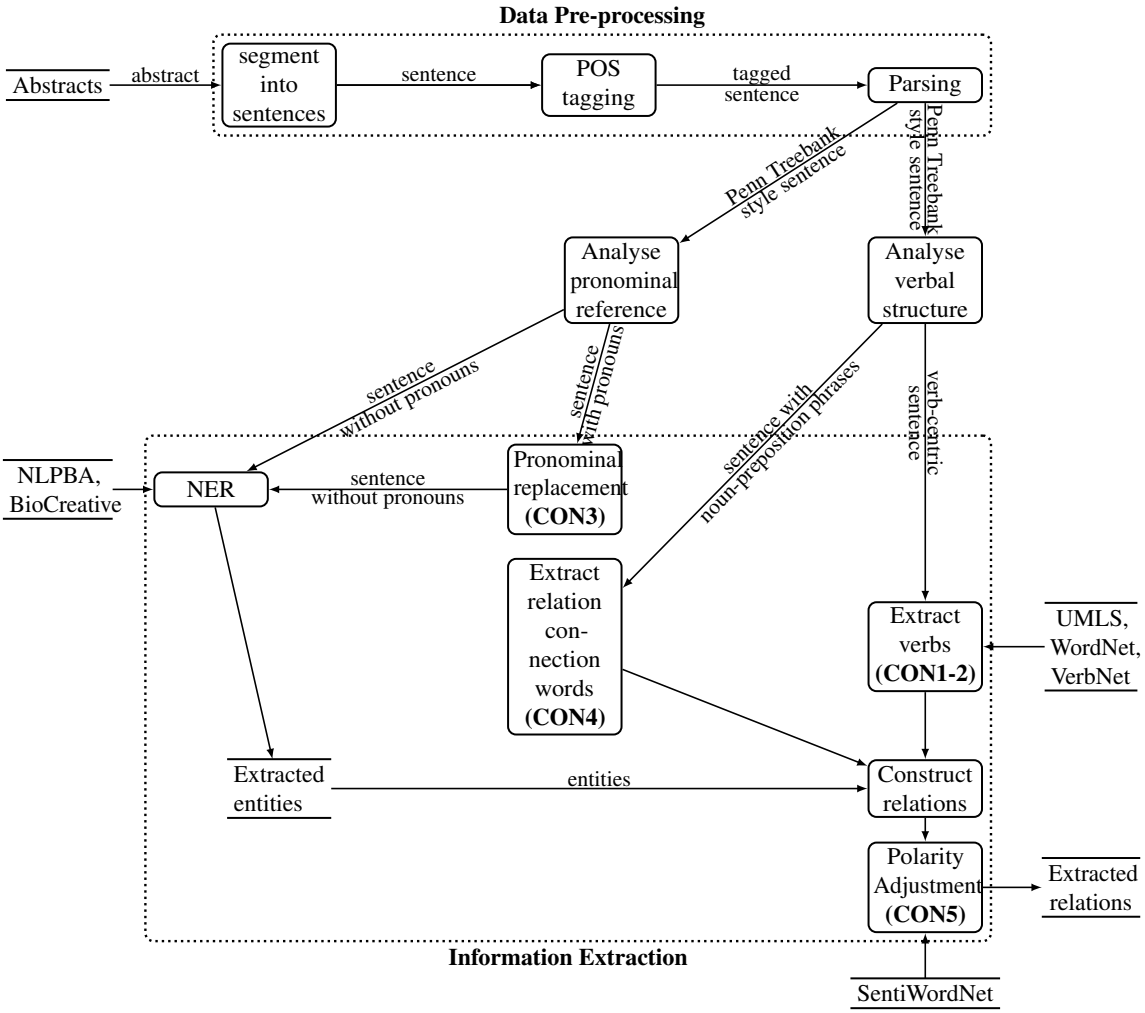


Fig. 4.1 Overview of a single iteration of the extraction process

SentiWordNet [Baccianella et al., 2010, Cambria et al., 2010]. Specifically, it identifies specific verbs, adjectives and adverbs modifying the nature of extracted relationships and assigns a *positive*, *negative* or *neutral* "polarity" to them. Extracted relationships with positive polarities are recorded for evaluation, and those with neutral or negative polarities are not. Finally, the system saves the resulting relation into a list of *Entity* | *relation connection* | *Entity* constructs.

An overview of the entire process is shown in Figure 4.1. The following sections explain those steps in detail.

4.2 Data Pre-processing

Given a document dataset, the first step is to perform data pre-processing using standard NLP techniques: each document is segmented into sentences, and each sentence is analysed, parsed and tagged (see “Data Pre-processing" in Figure 4.1). In this study, OpenNLP was used in an algorithm, written in R, for POS tagging and parsing sentences into structured extended markup language (XML) format. The algorithm detects the sentence boundaries, determines the lemma of each word in the sentences and labels each word with grammatical roles such as noun, verb, adjective, etc. Finally, the algorithm determines the structures of the sentences based on POS tags and dependencies. The algorithm output consists of sentences with the corresponding POS tagged words in Penn Treebank style, showing the grammatical constructs the sentence is composed of.

4.3 Entity Extraction

After pre-processing each sentence, the named entities are extracted using an existing Named Entity Recognition tool. We evaluated the performance of three NER techniques for our purposes: LingPipe, MetaMap and Abner. The results are presented in Section 4.6.

Table 4.1 shows a sample of the entities and their corresponding entity types that were abstracted from an article by means of Abner. That said, the approach presented herein allows for the NER tool to be substituted by another as circumstances demand. The performance of different NER tools may vary depending on the domain, so different applications may call for different NER tools.

Table 4.1 Example of extracted Entities

Candidate Entity	Entity Type
Fatty acid	Lipid
Vein graft	DNA
Cod liver	Body Part
Eicosapentaenoic acid	RNA
Cod-liver rich	cell Type

Pronominal References **(CON3)**

Human authors normally use a pronominal reference to express themselves more concisely. NER tools are unable to recognise such pronominal references. For example, in the sentence

> *"Magnesium is an essential micronutrient for human body, and its deficiency has been associated with risk of noncommunicable diseases."* [Hermes Sales et al., 2014]

the pronoun "its" refers to the entity "magnesium" in the clause sentence. Existing NER tools are unable to extract the entities "magnesium deficiency" because the reference to magnesium is implicit in the term "its deficiency".

Our algorithm deals with pronominal references by employing a co-reference resolution component proposed by Clark et al. in order to form chains between pronouns and their corresponding nominal words [Clark and Manning, 2016a,b]. We can use other existing co-reference resolution components to compute the associations, such as Stanford CoreNLP. The Clark's co-reference resolution component is only applied to sentences containing pronouns. Our algorithm uses the chains formed to replace the pronoun with its corresponding nominal word if the word was tagged as a bio-entity by the NER tool. This is described in Algorithm 3.

Algorithm 3 replaces most of the pronominal words with their corresponding bio-entities. The resulting sentences, where the pronouns have been appropriately replaced by entities, are then re-submitted to the NER tool for entity extraction.

```
Data: Penn Treebank style sentence with POS and NER tags
Result: sentence without pronouns
Split words;
if current sentence contains pronouns then
    if current sentence contains more than one entities then
        find the pronominal and nominal co-reference resolution;
        while more pronouns to process do
            if corresponding nominal word is tagged as a bio-entity then
                replace the pronoun with the corresponding entity;
            end
        end
    end
end
```

Algorithm 3: Algorithm for Pronominal Modification

Once the entities are fully extracted from the sentence, we move to the actual relationship extraction.

4.4 Relationship Extraction

Verb-centric algorithms normally identify potentially relevant sentences by analysing whether they contain a verb of interest. However, this approach ignores relationships embedded in

noun-preposition phrases. Therefore, the precision of the relationship extraction algorithm can be increased by enabling it to extract relationships expressed in the latter way.

Intuitively, our solution is simple. Since relationships involve at least two entities, we check if the sentence contains more than one recognised entity, instead of using the main verb as the main factor for selecting a relevant sentence. If a sentence contains two or more recognised entities, then it may embed a relationship between them. Next, the sentence structure is analysed and classified into two groups. In the first group, we include the sentences with a conventional verb-centric grammatical structure, which can be processed by a traditional verb-based algorithm. This is described in Section 4.4.1. In the second group, we include sentences with a more complex grammatical structure, but that nevertheless may contain an embedded relationship. We deal with these sentences using the technique described in Section 4.4.2.

4.4.1 Relationship extraction from verb-centric structures

Conventional verb-centric relationship extraction algorithms look for simple co-occurrences, where a biomedical verb appears between two entities. For example, a single-relation extraction algorithm can extract a simple co-occurrence relation *magnesium* | *influences* | *Biological Clock function* from the sentence

> *"The quality of magnesium status directly influences the Biological Clock function (BC)."* [Durlach et al., 2005]

Such co-occurrences are most commonly found in basic sentences of the form *Subject-Verb-Object*. The UMLS semantic network [Humphreys and Lindberg, 1993] contains 54 verbs or verb phrases that are commonly used to describe the relations that exist between biomedical entities. Using these 54 tokens, we can identify sentences describing relevant relationships. However, in order to make the relationship extraction more robust and expand the list of verbs that are deemed to potentially express a relationship of interest, we use WordNet [University, 2010] to expand the UMLS list so that it includes other verbs that are also semantically *similar*. For instance, "lower" is not from UMLS list, but if we use WordNet, we can extract the verb since it is semantic similar to the "decrease". Due to the fact that UMLS is focusing on biomedical domains, VerbNet [Schuler, 2005] is used to construct the initial verb list for datasets from general domains, such as 20Newsgroups. They are also expanded by WordNet to construct the final list of verbs.

Due to the fact that authors commonly use more complicated sentence structures, we developed Algorithm 4 to determining whether the sentence is a verb-centric structure sentence. It analyses the structure of a given sentence or semantic smaller unit. If the given

Data: Penn Treebank style sentence with POS tag and NER tags but without pronouns
Result: Semantic units with only one main verb and no noun phrases
Split words;
if *current sentence contains more than one recognised bio-entity* **then**
  if *current sentence contains only one main verb and no noun phrases* **then**
    Process sentence using Algorithm 5
  else
    if *current sentence matches* **(SS1)**-**(SS3) then**
      separate sentence into semantic units;
      foreach *semantic unit U in sentence* **do**
        Process *U* using Algorithm 5
      end
    end
  end
end

Algorithm 4: Algorithm for determining verb-centric structures

structure consists of only a single verb and no noun-phrases, then Algorithm 5 is applied to extract the main verb and any relationship it expresses through co-occurrence. Otherwise, Algorithm 4 aims to decompose the sentence into smaller semantic units based on known templates (SS1-SS3, explained below), each of which is analysed recursively by Algorithm 5.

The input of Algorithm 4 is a Penn Treebank style sentence with POS tags. The algorithm seeks to identify three types of structures **(SS1)–(SS3)**, which are commonly used by authors to express relationships. We illustrate structures **(SS1)–(SS3)** and how they are dealt with in Example 4.1. Sentences or semantic units of each of these structure types can be re-arranged into the atomic semantic units that each semantic unit only contains one main verb. Since the relative pronoun structure and conjunctive structure of the main sentence will have been recognised by the OpenNLP parser, the sentence can be partitioned based on the Penn Treebank style. If the relative pronoun occurs just behind a noun or a noun phrase, the algorithm will recognise the clause sentence as a smaller unit and assert the noun into the new unit. If the parent sub-tree of each conjunction corresponds to the entire sentence, the algorithm will recognise it as a sentence-level word and break the sentence into smaller semantic units. The re-arranged semantic unit can then be dealt with by the conventional co-occurrence Algorithm 5.

Example 4.1 (Processing complex sentence structures)**.**
(SS1) ***Clauses structures*** *are structures of the form* "... entity1 that/which verb ...entity2" *(using clauses to describe multiple verb-based relations in one sentence). For example,*

Data: Penn Treebank style semantic unit with POS tags
Result: Relation verbs
Split words;
if *current sentence contains at least one verb that is semantically similar to one of the verb list* **then**
  while *more words to process* **do**
    read current word;
    if *current word is the main verb* **then**
      add the current word to relation-verbs;
    end
    go to the next word;
  end
else
  exit and go to the next unit;
end

Algorithm 5: Algorithm for verb extraction

> "We propose that between attacks these metabolic shifts cause instability of neuronal function which enhances the susceptibility of brain to develop a migraine attack". [Welch and Ramadan, 1995]

There are two relation verbs, cause and enhances, which are connected by the relative pronoun word which. The relative pronoun "that" appears after the verb "propose", while "which" appears after the noun phrase "instability of neuronal function". Therefore, the sentence is divided into two parts at the relative pronoun "which", resulting in two smaller semantic units each containing an independent verb-based relation:

1. "We propose that between attacks these metabolic shifts cause instability of neuronal function."
2. "Instability of neuronal function enhances the susceptibility of brain to develop a migraine attack".

The sentence will produce two relationships shown in Table 4.2.
(SS2) ***Sentence level conjunctive structures:*** "...entity1 ...verb ...entity2 and/but verb ...entity3" *(using conjunctive structure to describe multiple verb-based relations in one sentence). For example,*

> *"Female hormones lower magnesium but increase calcium levels which enhance migraine ubiquitousness."* [Dhillon et al., 2011]

There are two relation verbs, lower and increase, which are connected by the conjunctive word but. The conjunction "but" has the entire sentence recognised as its parent sub-tree. Therefore, the sentence is divided into two parts at the conjunction but, resulting in two smaller semantic units each containing an independent verb-based relation. Considering **(SS1)** *together, three smaller semantic units can be obtained:*

1. "Female hormones lower magnesium."
2. "Female hormones increase calcium levels."
3. "Calcium levels enhance migraine ubiquitousness.

The sentence will produce three relationships shown in Table 4.3.
(SS3) ***Phrase level conjunctive structures:*** "…entity1 …verb …entity2, entity3, and entity4" *(using a conjunctive structure to describe a single verb-based many-to-many relationship). For example,*

> *"Low magnesium intakes and blood levels have been associated with type 2 diabetes, metabolic syndrome, elevated C reactive protein, hypertension, atherosclerotic vascular disease, sudden cardiac death, osteoporosis, migraine headache, asthma, and colon cancer."* [Rosanoff et al., 2012]

There is a two-to-ten relation. The two conjunctions "and" both return phrases as their parents. Therefore, the sentence will not be divided into smaller semantic units. The algorithm first considers all the entities located before the main verb as one entity and then breaks them apart. The same happens to the entities located after the main verb. Therefore, the above sentence will produce 20 relationships. Table 4.4 shows four of the extracted relationships.

Table 4.2 Example of extracted relations from **(SS1)** sentence.

Subject	Verb	Object
Metabolic shifts	cause	instability of neuronal function
Instability of neuronal function	enhances	migraine attack

Algorithm 5 ignores sentences with verbs not from the list and fails to extract relations. For example, no relationship is extracted from the sentence

> *"These results suggest a profound effect of the combined supply of Mg and Mn on the biosynthesis of terpenes and phenolics."* [Farzadfar et al., 2017]

Table 4.3 Example of extracted relations from **(SS2)** sentence.

Subject	Verb	Object
Female hormones	lower	magnesium
Female hormones	increase	calcium levels
Calcium levels	enhance	migraine ubiquitousness

Table 4.4 Four example of extracted relations from **(SS3)** sentence.

Subject	Verb	Object
Low magnesium intakes	been associated with	type 2 diabetes
Low magnesium intakes	been associated with	metabolic syndrome
Blood levels	been associated with	type 2 diabetes
Blood levels	been associated with	metabolic syndrome

because *"suggest"* is not semantically similar to any words from the UMLS list.

At this point, most of the main verbs in a sentence have been extracted, and we are now ready to construct the relations using the verbs and the entities previously recognised. The algorithm scans the positions of each term in the semantic unit and recalls the locations of the main verb and bio-entities. It then extracts the bio-entities that are located before and after the main verb. Algorithm 6 describes this process.

4.4.2 Dealing with Noun-Preposition Phrases

As we mentioned in Section 4.4, Algorithm 5 is not able to extract relationships expressed within noun-preposition phrases. In this section, we explain how some of the most common complex grammatical structures can be dealt with. In this work, we consider three kinds of noun-preposition phrases matching patterns **(NPP1-3)**.

After analysing a sentence's structure, if the sentence contains preposition words such as *between, of* and *due to*, we consider them as sentences with noun-preposition phrases and do not use Algorithm 5 to extract verbs. Instead, we deal with them using rules to extract relation connection words. Example 4.2 shows the structures of patterns **(NPP1-3)** and how these rules are used to help extract relationships from noun-preposition phrases.

Data: Penn Treebank style semantic unit with POS tags
Result: Entities and relation verbs
Split words;
Scan each word and remember its position;
Construct relation from main verb;
while *not the end of the semantic unit* **do**
  if *current word is a bio-entity* **then**
    if *it appears before the main verb* **then**
      add the current word as subject of relation;
    else
      add the current word as an object of relation;
    end
  end
  go to the next word;
end

Algorithm 6: Algorithm for constructing relations

Example 4.2 (Dealing with noun-preposition phrases)**.**
(NPP1) " effect/influence/... between entity-A and entity-B"*: The algorithm identifies the entity after* between *and before* and *(entity-A) as the left entity, the one immediately after* and *(entity-B) as the right entity. The relation connection words are the main verb of the sentence plus all the words between the main verb and the word* between*. For example, the sentence*

> "The objective of this study was to determine the effect between vitamin D status and broad gene expression in healthy adults." [Hossein-Nezhad et al., 2013]

contains a relationship between Vitamin D status *and* gene expression *connected by a noun-preposition phrase* "between ... and ...". *The entity* "Vitamin D status" *appears after* between *and before* and, *and the entity* " gene expression" *appears immediately after* and. *Therefore, the extracted relationship is* Vitamin D status | studied the positive influences | gene expression.
(NPP2) "effect/influence/... of entity-A on entity-B"*: The algorithm identifies the entity after* of *and before* on *(entity-A) as the left entity, the one immediately after* on *(entity-B) as the right entity. The relation connection words are the main verb of the sentence plus all the words between the main verb and the word* of*. For example, the sentence*

> *"These results suggest a profound effect of the combined supply of Mg and Mn on the biosynthesis of terpenes and phenolics."* [Farzadfar et al., 2017]

contains a relationship between Mg and Mn *and* biosynthesis of terpenes and phenolics *connected by a noun-preposition phrase* "of ... on...". *The entity* "Mg and Mn" *appears after*

of *and before* on, *and the entity* "biosynthesis of terpenes and phenolics" *appears immediately after* on. *Therefore, the extracted relationship is* Mg and Mn | suggest a profound effect | biosynthesis of terpenes and phenolics.
(NPP3) "entity-A due to entity-B"*: The algorithm identifies the entity before* due to *(entity-A) as the right entity, the one after* due to *(entity-B) as the left entity. The relation connection words are* due to *indicating the cause-effect relations. For example, the sentence*

> *"The lesion was then classified as extra-gonadal yolk sac tumor due to alarming ultrasound features, later confirmed at MRI and pathology."* [Esposito et al., 2016]

contains a relationship between extra-gonadal yolk sac tumor *and* ultrasound features *connected by a preposition phrase* "due to". *The entity* "extra-gonadal yolk sac tumor" *appears before* due to *and the entity* "ultrasound features" *appears after* due to. *Therefore, the extracted relationship is* extra-gonadal yolk sac tumor | due to | ultrasound features.

The addition of these three rules above means that our algorithm is ready to deal not only with verb-centric relationships but also with noun-preposition phrases relationships matching patterns **(NPP1)**-**(NPP3)** above and extract their relation connection words (verbs and noun-preposition phrases). We are now ready to construct relations with the identified entities and relation connection words (*RCW*) of interest using either Algorithm 6 or the rules explained above. However, we do not want to record all extracted relationships from sentences containing noun-preposition phrases. It is not uncommon for authors to mention a relationship without asserting it. Therefore, we analyse the extracted relation connection words to determine whether to record an extracted relationship in the following section.

4.5 Polarity Adjustment

In our experiments, we noticed that most false positives occurred when a relationship is mentioned but not necessarily asserted. For instance, the sentence

> *1) "The purpose of this study was to analyse the significance of level IIB dissection in patients of oral cavity cancer who underwent primary surgery with functional neck dissection."* [Chheda et al., 2017]

does not assert the existence of a relationship between level IIB dissection and oral cavity cancer but merely describes the objective of the study itself. A conventional algorithm would extract a relationship *level IIB dissection* | *analyse the significance* | *oral cavity cancer* while we want to avoid that.

On the other hand, some sentences express a relationship in the opposite way that the verb alone suggests. For example

> *2) "These results reduce the possibilities of inhibitory interactions between the mutant and wild-type ChlI1a and ChlI1b proteins".* [Campbell et al., 2015a]

In this example, the relationship is expressed through the use of the negative verb expression *"reduce the possibilities."* A conventional algorithm would still extract the relationship between *"mutant ChlI1a and ChlI1b"* and *"wild-type ChlI1a and ChlI1b"* while we want to avoid that.

These scenarios normally happen in relationships expressed within noun-preposition phrases. In order to deal with these special cases, we introduce the concept of *relationship polarity* that can be used to aid in the relationship extraction. When we extract a relationship, we associate with it one of the three possible polarities: positive, negative or neutral. A positive polarity suggests the existence of a relationship; a negative polarity denies the existence of a relationship; and a neutral polarity indicates a reference to a relationship, without asserting or denying its existence. The relationship in our example *1)* would be classified as neutral and the relationship and in our example *2)* would be classified as negative. Although we extract relationships with either of the three polarities, we only record those with positive polarities. We keep others with neutral/negative polarities for future uses.

In this work, we focus on determining the polarities of relationships embedded in sentences containing Noun-Preposition Phrases. The algorithm takes the relation connection words discovered in those structures as input and determines how each relation connection word affects the position that expressed about that relationship. Specifically, for each relation connection work, a polarity score is computed, where positive numbers indicate that the associated relationship is asserted, negative numbers indicate that the relationship is rejected and numbers close to 0 indicate that neutral position is taken with respect to the relationship.

Our approach uses SentiWordNet, an enhanced lexical resource for sentiment analysis and opinion monitoring. SentiWordNet has assigned over 120 thousand verbs, adjectives and adverbs from WordNet two sentiment scores: PosScore and NegScore. PosScore represents the positive polarity of a word, while NegScore represents the negative polarity of a word [Baccianella et al., 2010, Cambria et al., 2010].

The polarity score for a relation connection word *RCW* is computed as follows:

$$\text{PolScore}(RCW) = \text{PosScore}(RCW) - \text{NegScore}(RCW) \tag{4.1}$$

Table 4.5 shows some examples of PolScores extracted from SentiWordNet.

Table 4.5 Examples words with their PosScore, NegScore and PolScore

Words	PosScore	NegScore	PolScore
Suggest	0.5	0	0.5
Verify	0.5	0	0.5
Deny	0	0.875	-0.875
Reduce	0	0.25	-0.25
Assess	0	0	0
Analyze	0	0	0
Profound	0.375	0	0.375
Difficult	0	0.75	-0.75

To compute the polarity of a relationship in a phrase containing a set of relation connection words P, the polarity scores of the relation connection words in that phrase are added together:

$$\text{PolScore}(P) = \sum_{RCW \in P} \text{PolScore}(RCW) \tag{4.2}$$

If the polarity score of the phrase is larger than 0, the extracted relations will be recorded. Otherwise, they will not be recorded. The results of the algorithm can be explained with a number of examples. The algorithm will not record any relationship from the sentence

> *1) "The purpose of this study was to analyze the significance of level IIB dissection in patients of oral cavity cancer who underwent primary surgery with functional neck dissection."* [Chheda et al., 2017]

Because the PolScore of the relation connection word *analyse* is 0. Similar, the algorithm will not record any relationships from the sentence

> *2) "These results reduce the possibilities of inhibitory interactions between the mutant and wild-type ChlI1a and ChlI1b proteins"*. [Campbell et al., 2015a]

In this example, the relationship is expressed through the use of the negative verb expression "*reduce the possibilities.*" and the PolScore of the relation connection word *reduce* is -0.125. However, the relation will be extracted and recorded from the sentence:

> *"These results suggest a profound effect of the combined supply of Mg and Mn on the biosynthesis of terpenes and phenolics."* [Farzadfar et al., 2017]

because PolScore("suggest") + PolScore("profound") = 0.875.

This section has presented a number of techniques that aim to improve verb-based relationship extraction techniques. In the next section, the effect of these techniques will be assessed.

4.6 Evaluation Experiments

In this section, we first evaluate the performances of three NER tools in the biomedical domain. Then we evaluate the effectiveness of the proposed improvements in identifying relationships in both biomedical text and daily life email texts using two different datasets.

4.6.1 Experimental Setup

To perform a comparative analysis, two datasets (presented below) were created and used: one comes from the biomedical domain, and the other is from 20 Newsgroups dataset. The obtained NER tools and relation extraction algorithms were evaluated on these two datasets.

For the purpose of evaluating the NER tools by means of F-measures, a *true positive* (TP) represents an entity that has been correctly identified by the NER, a *false positive* (FP) represents an entity that has been incorrectly identified (i.e., it should not have been extracted); and a *false negative* (FN) represents an entity that should have been extracted but was missed by the NER.

For the purpose of evaluating the relation extraction algorithms by means of F-measures, a *true positive* (TP) represents a relation that has been correctly identified by the extraction algorithm, a *false positive* (FP) represents a relation that has been incorrectly identified (i.e., it should not have been extracted); and a *false negative* (FN) represents a relation that should have been extracted but was missed by the extraction algorithm.

Using these definitions for true/false positives/negatives, precision and recall are defined as in Equation 2.1, 2.2 and 2.3.

All experiments were performed on a PC with an i7 processor, an NVIDIA GeForce GPU GTX 970M graphics card, and 16GB RAM.

4.6.2 Datasets Used in the Analysis

For the purpose of the evaluation, we created a dataset consisting of 600 sentences for the biomedical domain from papers taken from PubMed that we call PubMed600 dataset. This dataset is publicly available in Github and can be downloaded from https://github.com/qihao71/PubMed-dataset. We also used 20 Newsgroups dataset for the general domain. Here we briefly describe how each of these datasets was used.

PubMed600 dataset

PubMed consists of more than 26 million publications from the MEDLINE bibliographic database, life science journals, and online books. PubMed also includes the full text of the

Table 4.6 Sample of the downloaded text data from PubMed600 dataset

PMID	Title	Abstract
26730018	"A Novel Surgical Technique for Thyroid Cancer with Intra-Cricotracheal Invasion: Windmill Resection and Tetris Reconstruction"	"The most effective treatment for thyroid cancer (TC) invading into the larynx and trachea is complete surgical resection of the tumor, but currently employed techniques are less than ideal. We report a novel surgical technique, which we named Windmill resection and Tetris reconstruction, for patients with TC invading into the laryngeal lumen. We treated eight cases of TC with invasion into the laryngeal lumen by Windmill resection and Tetris reconstruction. [...]"
25400410	"Gellan gum-based mucoadhesive microspheres of almotriptan for nasal administration: Formulation optimisation using factorial design, characterisation, and in vitro evaluation"	"Almotriptan malate (ALM), indicated for the treatment of migraine in adults is not a drug candidate feasible to be administered through the oral route during the attack due to its associated symptoms such as nausea and vomiting. This obviates an alternative dosage form and nasal drug delivery is a good substitute to oral and parenteral administration. [...]"

biomedical articles, including their abstracts. Each record contains a PubMed ID (PMID), the title of the article and its abstract in plain text format, ready to be analysed by NLP tools. As in others existing work on relationship extraction from biomedical texts, we consider only the abstracts of the articles because they are consistently available and contain a representative summary of the main text [Sharma et al., 2010, Feldman et al., 2002, Kim et al., 2006].

Methodology

We first randomly selected 362 abstracts about "magnesium deficiency", "migraine attack" and "cancer" as keyword search from PubMed. In order to compare the performance achieved by our algorithms with previous results by other works, due to the reason that the list of articles used in previous works comprising the dataset used in their evaluation was not publicised, we decided to use the same topics: "cancer", "magnesium deficiency", "migraine attack" and to keep the comparison as close as possible to the other evaluation [Sharma et al., 2010, Khordad and Mercer, 2017]. A typical abstract contained roughly 8-10 sentences, and the dataset contains approximately 3000 sentences. Samples of the text data downloaded are shown in Table 4.6.

From these 362 abstracts, we construct the PubMed600 dataset by randomly selecting 600 sentences. These sentences were manually annotated by a biochemistry graduate student and us. 1545 biomedical named entities were annotated, including protein, DNA, RNA, body part and cell type. To define whether a word is a biomedical entity, UMLS were used like a

dictionary. 996 relationships between these biomedical named entities were annotated. For these 600 sentences, the distribution of the PubMed600 dataset is illustrated in Table 4.7.

Table 4.7 Distribution of the PubMed600 benchmark dataset

Sentence structures	Number of sentences
Co-references	209
(SS1)	121
(SS2)	135
(SS3)	154
(NPP1)	43
(NPP2)	37
(NPP3)	3

20 Newsgroups

As mentioned in section 3.4.2, 20 Newsgroups dataset is a commonly used benchmark dataset for various NLP tasks such as topic classification and relation extraction. From this dataset, Wang et al. created a benchmark dataset specifically for entity recognition and relation extraction [Wang et al., 2016]. They first selected 200 documents from 20 newsgroups dataset, i.e., 10 documents from each category. Then these documents were split into sentences. 3,232 sentences were randomly selected by them to construct this dataset. In this work, we evaluated our algorithm using this dataset.

4.6.3 Experimental Results

We discuss the results of the evaluation of three NER tools and relation extraction algorithms in the following part.

Evaluation of NER tools

For evaluating the three different NER tools in the biomedical domain, 100 sentences were randomly selected from the PubMed600 benchmark dataset. We conducted two rounds of experiments with these 100 sentences. Table 4.8 shows the average precision, recall and F-measures. In our dataset, Abner achieved the best results in each criterion and, therefore, we decided to use Abner as the main NER tool in the following relation extraction algorithm experiments.

Table 4.8 Evaluation results of NER tools

NER Tool	Precision	Recall	F-measure
LingPipe	0.687	0.622	0.653
MetaMap	0.635	0.581	0.607
Abner	0.714	0.640	0.675

Evaluation of relation extraction algorithms

We evaluated the performance of the entity-based algorithm in terms of precision, recall and F-measure on two datasets using as baseline the conventional rule-based algorithm (which only extracts a single *Entity* | *Verb* | *Entity* relation from each sentence). For PubMed600 dataset, Abner NER tool was employed. And for 20 Newsgroups dataset, StanfordNLP NER tool was employed. We also evaluated the performance of our entity-based algorithm with contributions **(CON1,2,3,4,5)** in different combinations on the same datasets.

For both datasets, standard NLP techniques were used to obtain separated Penn Treebank style sentences with POS tags from the dataset. Then, each Penn Treebank style sentence was analysed to check for pronominal references and sentences with pronouns were processed by Algorithm 3 for pronominal replacement. Subsequently, sentences without pronouns were fed into Abner to automatically recognise bio-entities and output the extracted entities. Next, the algorithm also took the Penn Treebank style sentences as input for analysing verbal structure. The verb-centric sentences were fed into Algorithm 4 to be matched against **(SS1)**-**(SS3)** and were fed into Algorithm 5 to extract main verbs. For the PubMed600 dataset, the list of main verbs consists of 54 verbs from UMLS and expanded by their semantically similar verbs from WordNet. For 20Newsgroups dataset, the list of main verbs consists of 113 verbs from VerbNet and expanded by their semantically similar verbs from WordNet. Sentences with noun-preposition phrases matching pattern **(NPP1)**-**(NPP3)** were processed to extract relation connection words based on rules. Then, extracted relations were constructed with the identified entities and relation connection words of interest using either Algorithm 6 or the rules explained above. The polarities of the extracted relations were then determined, and those that were positive were recorded. The application of all these steps incorporates contributions **(CON1,2,3,4,5)** and constitutes our entity-based algorithm.

Table 4.9 contains the evaluation results of all valid combinations of contributions **(CON1)**, **(CON2)**, **(CON3)**, **(CON4)** and **(CON5)**. Since **(CON5)** cannot be used without **(CON4)**, we have excluded the invalid combinations that include **(CON5)** without **(CON4)**. This results in 24 possible valid combinations. These combinations are represented as the sequence of digit 12345. A bar over the digit X indicates that **(CONX)** was not used in the combination. Therefore, $\overline{12345}$ represents a conventional rule-based algorithm without

Table 4.9 Results of valid combinations of contributions

	PubMed600 Dataset						20 Newsgroups Dataset					
Algorithm	TP	FP	FN	Precision	Recall	F-score	TP	FP	FN	Precision	Recall	F-score
$\overline{1}2345$	310	118	172	0.724	0.643	0.681	342	101	157	0.772	0.685	0.723
$\overline{1}2\overline{3}45$	323	110	167	0.746	0.659	0.700	354	94	152	0.790	0.700	0.742
$\overline{1}\overline{2}345$	328	108	164	0.752	0.667	0.707	350	88	162	0.799	0.684	0.737
$\overline{1}2\overline{3}\overline{4}5$	331	107	162	0.756	0.671	0.711	357	89	154	0.800	0.699	0.746
$\overline{1}2\overline{3}4\overline{5}$	345	104	151	0.768	0.696	0.730	368	85	147	0.812	0.715	0.760
$\overline{1}\overline{2}345$	351	99	150	0.780	0.701	0.738	372	79	149	0.825	0.714	0.765
$\overline{1}2\overline{3}\overline{4}\overline{5}$	369	93	147	0.799	0.715	0.755	398	75	127	0.841	0.758	0.797
$\overline{1}2\overline{3}4\overline{5}$	381	80	139	0.826	0.733	0.777	413	69	118	0.857	0.778	0.816
$\overline{1}2\overline{3}45$	401	76	123	0.841	0.765	0.801	421	59	120	0.877	0.779	0.825
$\overline{1}2345$	384	82	134	0.824	0.741	0.780	415	71	114	0.854	0.784	0.818
$\overline{1}2345$	407	69	124	0.855	0.766	0.808	427	60	113	0.877	0.791	0.832
$\overline{1}2345$	415	67	118	0.861	0.779	0.818	436	54	110	0.890	0.799	0.842
$1\overline{2}\overline{3}45$	384	81	135	0.826	0.740	0.780	419	68	113	0.860	0.788	0.822
$1\overline{2}\overline{3}4\overline{5}$	408	76	116	0.843	0.779	0.810	431	57	112	0.883	0.794	0.836
$1\overline{2}\overline{3}45$	428	59	113	0.879	0.791	0.833	458	45	97	0.911	0.825	0.866
$1\overline{2}3\overline{4}5$	412	71	117	0.853	0.779	0.814	439	53	108	0.892	0.803	0.846
$1\overline{2}3\overline{4}\overline{5}$	427	68	105	0.863	0.803	0.832	456	50	94	0.901	0.829	0.864
$1\overline{2}345$	439	60	101	0.880	0.813	0.845	464	41	95	0.919	0.830	0.873
$12\overline{3}\overline{4}5$	443	58	99	0.884	0.817	0.849	472	38	90	0.908	0.840	0.873
$12\overline{3}4\overline{5}$	466	56	78	0.893	0.857	0.874	490	37	73	0.93	0.870	0.899
$12\overline{3}45$	494	51	55	0.906	0.900	0.903	523	35	42	0.932	0.926	0.929
$123\overline{4}\overline{5}$	469	55	76	0.895	0.861	0.877	494	36	70	0.932	0.876	0.903
$1234\overline{5}$	502	53	45	0.905	0.918	0.911	531	33	36	0.941	0.937	0.939
12345	518	49	33	0.914	0.940	0.927	549	30	21	0.948	0.963	0.955

any of the contributions proposed in this paper. At the other extreme, 12345 represents the entity-based algorithm resulting from the enhancement of the conventional rule-based algorithm with all five contributions proposed in this chapter. Table 4.9 shows that the proposed entity-based algorithm achieves good accuracy in both the biomedical domain and daily life domain. Experiments with 20 Newsgroups dataset outperforms that with PubMed600 dataset since biomedical texts are more complicated than daily life texts.

The chart shown in Figure 4.2 visualises the results for nine key combinations examining the incremental benefit of each of the 5 contributions using PubMed600 Dataset. The results demonstrate that **(CON1)** and **(CON2)** each increase precision and recall of the baseline algorithm, as well as in combination with one another (**(CON1,2)**) [Hao et al., 2017].

Table 4.9 and Figure 4.2 also report the effect of extending **(CON1,2)** with the three new contributions **(CON3)**, **(CON4)** and **(CON5)** in different combinations. Note that **(CON5)** performs polarity adjustment on the relationships extracted from noun-preposition phrases by means of **(CON4)**. As such, **(CON5)** is only considered in combination with **(CON4)**. As shown in Table 4.9 and Figure 4.2, each of these contributions also improves both precision and recall. The marginal improvements in precision are reducing with contributions

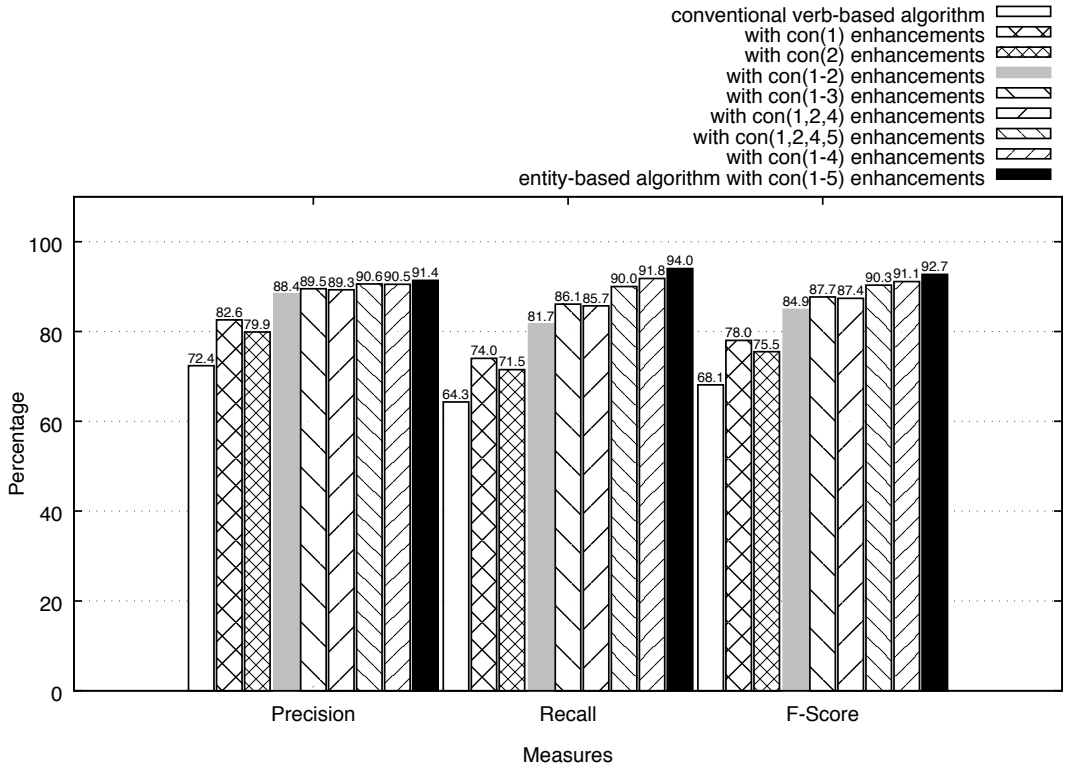


Fig. 4.2 Performance on the PubMed600 dataset

(CON3), **(CON4)** and **(CON5)**. This is not unexpected as precision achieved with **(CON1,2)** was already high and contributions **(CON3)**, **(CON4)** and **(CON5)** are primarily aimed at reducing false negatives. Consequently, recall improvements achieved by **(CON3)**, **(CON4)** and **(CON5)** on top of **(CON1,2)** are substantial. The entity-based algorithm with all contributions **(CON1,2,3,4,5)** achieved an overall precision of 0.914 with 0.94 recall in PubMed600 dataset and overall precision of 0.948 with 0.963 recall in 20 Newsgroups dataset, thereby offering significant advantages over the previous relationship extraction methods of this type.

Comparing with state-of-the-art work

Our proposed relation extraction algorithm with all contributions **(CON1,2,3,4,5)** are referred as entity-based algorithm. Table 4.10 shows the comparison results of our entity-based algorithm with some state-of-the-art work. For the biomedical dataset PubMed600, our algorithm achieves better results compared to the existing rule-based algorithm by 5-15 percentual points [Sharma et al., 2010, Khordad and Mercer, 2017]. For the general domain dataset 20 Newsgroups, our entity-based algorithm also achieves better results by 4 percentual points [Wang et al., 2016]. Comparing to state-of-the-art joint entity-relation extraction

Table 4.10 Experiment results of state-of-the-art work

Dataset	Algorithm	F1-score
PubMed600	[Sharma et al., 2010]	0.8736
	[Khordad and Mercer, 2017]	0.7777
	Entity-based algorithm	**0.927**
20 Newsgroups	[Wang et al., 2016]	0.916
	Entity-based algorithm	**0.955**
CoNLL04	[Bekoulis et al., 2018]	0.8049
	[Eberts and Ulges, 2019]	0.7147

models [Bekoulis et al., 2018], ours achieves better relation extraction results. Our algorithm also outperforms the state-of-the-art algorithm based on Transformers [Eberts and Ulges, 2019].

Impact on speed of (CON1,2,3,4,5)

Table 4.11 summarises the impact on speed of **(CON1,2,3,4,5)**. As expected, the conventional rule-based algorithm without **(CON1,2,3,4,5)** is the fastest for both datasets. **(CON3)** requires much more time to perform (an addition of 1 second/sentence) since this step relies on external toolkit to find co-reference resolution. The other four **(CON1,2,4,5)** requires nearly the same time to perform (average of 0.5 seconds/sentence). To sum up, our proposed entity-based algorithm with five extensions **(CON1,2,3,4,5)** only requires 2.5 times of time when without. This is negligible considering the huge improvements they achieved.

4.7 Discussion

Although our enhancements greatly improved the overall performance of the relationship extraction, our evaluation showed a number of situations that gave rise to false positives and false negatives.

Most false positives were caused by two common issues: the inability to deal with more complex prepositional phrases and the difficulty in understanding the use of adjective phrases associated with the relationships. For example, the sentence

> *"The combination of apricoxib and IL-27 resulted in augmentation of STAT1 activation."* [Lee et al., 2014]

contains a relationship *the combination of apricoxib and IL-27| resulted in |STAT1 activation*. However, the algorithm ignored the preposition phrase "the combination of" and extracted

Table 4.11 Average time required in seconds to process one sentence per algorithm

Algorithm	PubMed600 Dataset	20 Newsgroups Dataset
$\overline{12345}$	2.10	1.98
$\overline{1234}\overline{5}$	2.60	2.46
$\overline{123}\overline{45}$	3.11	2.74
$\overline{123}\overline{45}$	3.05	2.97
$\overline{123}\overline{4}\overline{5}$	3.64	3.36
$\overline{12}\overline{345}$	4.13	3.93
$\overline{1}\overline{2345}$	2.58	2.34
$\overline{1}\overline{23}\overline{45}$	3.08	2.81
$\overline{1}\overline{23}\overline{45}$	3.51	3.22
$\overline{1}\overline{2345}$	3.67	3.39
$\overline{1}\overline{234}\overline{5}$	4.17	3.85
$\overline{1}2345$	4.71	4.65
$1\overline{2345}$	2.59	2.37
$1\overline{23}\overline{4}\overline{5}$	3.07	2.78
$1\overline{23}\overline{45}$	3.55	3.30
$1\overline{23}\overline{45}$	3.51	3.21
$1\overline{2}\overline{34}\overline{5}$	4.29	4.08
$1\overline{2}345$	4.74	4.57
$12\overline{345}$	3.13	2.96
$12\overline{3}\overline{4}\overline{5}$	3.56	3.32
$12\overline{3}45$	4.29	4.01
$1234\overline{5}$	4.11	3.99
$1234\overline{5}$	4.72	4.51
12345	5.32	4.91

two relationships *apricoxib| resulted in |STAT1 activation* and *IL-27| resulted in |STAT1 activation*. An example of the second type is the sentence

> *"Establishing the relationship between glaucoma and headaches is a formidable challenge."* [Lipton et al., 2014]

The sentence does *not* assert the existence of a relationship between glaucoma and headaches because this is part of the subject and the word "challenge" actually implies that identifying the relationship is difficult. However, our algorithm is not yet sophisticated enough to deal with this scenario and extracted the relationship.

Some of the false negatives during entity recognition were caused by issues with the co-reference resolution module. For example, the sentence

> *"The aim of this study was to discern whether a relation between biochemical parameters, sonography and musculoskeletal data exists in cases of hyperthyroidism and whether they are modifiable through supplementation with selenomethionine and magnesium citrate as well as by acupuncture and manual medicine methods."* [Moncayo and Moncayo, 2015]

contains too many entities occurring before the pronoun "they". The system failed to identify the corresponding nominal words referred by the pronominal word. Therefore our algorithm failed to extract the correct entities embedded in the sentence.

In addition, some of the false negatives were caused by the inability to deal with references to entities occurring outside the sentence. For example, the sentence

> *"It is generally well tolerated and has excellent oral bioavailability, providing significant benefit in the treatment of invasive fungal infections."* [Willis et al., 2014]

contains a pronoun "it" referring to an entity occurring in a *previous* sentence, but our algorithm is currently only able to deal with a single sentence at a time and therefore cannot yet handle co-references occurring across multiple sentences. Another example is the sentence

> *"The supplementation brought a reduction of the vascularisation indices and reduced the incidence of idiopathic moving toes."* [Moncayo and Moncayo, 2015]

This sentence actually contains two relationships: *the supplementation (of an entity)* | *reduced* | *the vascularisation indices* and *the supplementation (of an entity)* | *reduced* | *idiopathic moving toes*. However, the algorithm failed to extract these relationships due to the absence of the entity's name in this single sentence.

4.8 Summary

Automatic relationship extraction from unstructured data written in natural language is an important field of computational linguistic research and has recently gained a lot of attention in the literature. Currently, the most efficient approaches for relationship extraction are based on machine learning or use special rules identifying patterns in the text describing relationships between entities. However, the machine learning approaches used in this field are supervised and require manually annotated training data. Similarly, conventional rule-based approaches require domain experts to generate domain rules for pattern matching.

These can be costly. Researchers are trying to improve conventional rule-based approaches to retain a good level of precision while requiring minimal human intervention. Most of them focus on algorithms that extract relationships by scanning sentences for verbs and then analysing whether they are associated with entities in the domain of interest.

Existing rule-based approaches focusing on verbs are not without limitations. They search exclusively for relations expressed using a *Entity-Verb-Entity* pattern. This limits them to identifying a single relation in each sentence. Moreover, many relations do not follow this pattern. In particular, verb-based approaches cannot extract relationships expressed within noun-phrases, or relationships where an entity is referred to by a pronoun. Improving and expanding on the enhanced verb-based algorithm presented in [Hao et al., 2017], the entity-based algorithm proposed in this chapter can now address all of these shortcomings, achieving better overall results.

The overall process can be summarised as follows. Standard NLP techniques are used to analyse and parse the grammatical structure of a sentence written in English. Then, the sentence's components are tagged and its pronouns replaced by the entities they refer to. Subsequently, unlike in conventional rule-based approaches, potential relationships are identified not by the recognition of the pattern *Entity-Verb-Entity*, but by the existence of multiple entities within the same sentence. Because of this, we called the enhanced algorithm *entity-based*. Now, using the grammatical structure of the sentence, several transformations are performed, allowing the extraction of relationships embedded within complex structures, including clauses, conjunctions, and noun-preposition phrases. Finally, we introduced the new concept of *relationship polarity*, which adjusts the extraction of the relationship so that it takes into account adjectives and adverbs modifying its intended meaning.

We evaluated the performance of this new approach on two datasets, one from the biomedical domain consisting of 600 sentences and one from general domain consisting of 600 sentences. In order to further distinguish the improvement achieved by each contribution, we analysed the extraction results of algorithms using different combinations of contributions **(CON1)**, **(CON2)**, **(CON3)**, **(CON4)** and **(CON5)**. The complete set of results is shown in Table 4.9. The conventional rule-based algorithm without any improvements achieves an overall precision of 0.724 and had 0.643 recall in PubMed600 dataset and overall precision of 0.772 with 0.685 recall in 20 Newsgroups dataset. The addition of contributions **(CON1)** and **(CON2)** each resulted in a substantial increase in precision and recall. In combination, **(CON1,2)** yields an overall precision of 0.884 and recall of 0.817 in PubMed600 dataset and overall precision of 0.908 with 0.84 recall in 20 Newsgroups dataset. These results corroborate the evaluation results reported in [Hao et al., 2017]. We also compared the results of extending **(CON1,2)** with the three new contributions **(CON3)**, **(CON4)** and **(CON5)** in

different combinations. Each those combinations offered improvements in both precision and recall. Our evaluation shows that **(CON3)**, **(CON4)** and **(CON5)** benefit recall primarily. The entity-based algorithm with contributions **(CON1,2,3,4,5)** achieved an overall precision of 0.914 with 0.94 recall in PubMed600 dataset and overall precision of 0.948 with 0.963 recall in 20 Newsgroups dataset, thereby offering significant advantages over the previous works. These results show that the shortcomings of conventional verb-based algorithms can be addressed by the techniques proposed herein. With both datasets, the proposed entity-based algorithm achieves a stable accuracy, which means it can be easily applied in different domains by using different training corpus and NER tools. (see Table 4.9 and Figure 4.2 for a detailed comparison).

At this point, the extracted relationship information is ready to be included in the topic model. In the following chapters, we focus on incorporating relations between words into the OLDA topic model to classify text documents based on their contents and topics.

Chapter 5

Ontology Driven Topic Classification with Structured Relationships

In this chapter, we further improve the OLDA topic model (described in Chapter 3) by incorporating relationships between words and ontology concepts.

As described in the previous chapter, by including an ontological knowledge base as an intermediate concept component in LDA, OLDA allows topics to be defined more generally by ontological concepts instead of words so that the semantical meaning of words can be captured. However, the ontological knowledge introduced may contain irrelevant information in the context. For example, the following sentences from 20 Newsgroups dataset are about three different topics.

Example 5.1. Topic(*comp.sys.ibm.pi.hardware*): "IBM launched a new Windows laptop. A lot of money is invested for this laptop."

Example 5.2. Topic(*misc.forsale*): "David broke my car windows yesterday. He has to sell his laptop to pay for the repairs."

Example 5.3. Topic(*soc.religion.christian*): "David purchased a new Dell laptop as a Christmas gift."

However, LDA would classify them into the same topic category since they contain same words *"laptop", "windows"*. OLDA would also classifies them incorrectly because these sentences contain different words with same ontological concepts: word *"IBM"* in Example 5.1 and word *"Dell"* in Example 5.3 have the same ontological concept *"company"*; They also contain same words with different semantical meanings in different context: word *"windows"* in Example 5.1 means *"software* while it in Example 5.2 means *"barrier"*. In order to only consider the relevant concepts in the context, we incorporate the relationships between

words and ontology concepts embedded in each document so that the semantical structures in texts can be considered. This can be done by extracting relationships between words embedded in the texts using algorithms described in Chapter 4. Then relationships between concepts that are associated with corresponding words are constructed using ontological knowledge. Our approach uses these relationships to find implicit knowledge in the context and therefore increase the overall accuracy of the topic modelling and classification. In our previous example, the three texts would be classified into different topics because they do not share the same relationships between words or concepts. This has the following advantages: (i) it allows the topics to be defined more specifically in terms of contextual relationships between words and concepts rather than general concepts, and this captures the contextual semantical meaning of words more accurately; (ii) as a side-effect, we will see that this extra dimension helps to reduce the training and classification times. In virtue of the use of this relationship knowledge, we call the resulting technique *Relation-Ontology Driven LDA for Topic Classification* (ROLDA).

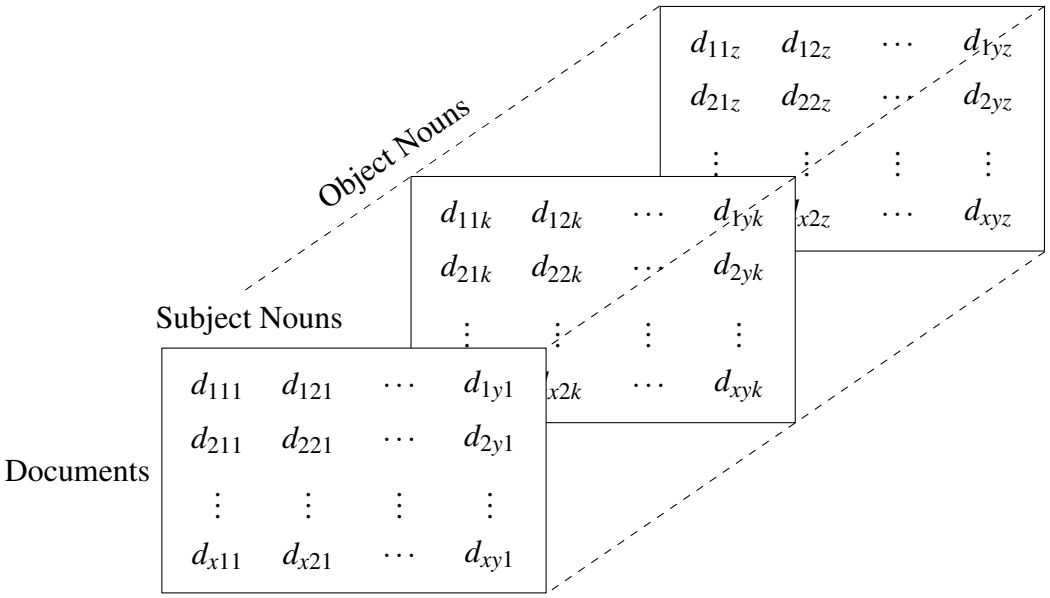
ROLDA can also slightly reduce the training and classification times compared to OLDA by incorporating the extra relation dimension. As for OLDA, ROLDA can also employ a self-training phase. The resulting technique ST-ROLDA can further reduce the training and classification times and reduce the amount of manually classified training data.

In order to further speed up the training process, a distributed cloud computing process was developed that can be used for OLDA, ROLDA and any other topic modelling approaches. It significantly reduces the training time by nearly half.

The remainder of this chapter is organised as follows. Section 5.1 describes the methodology of ROLDA. Section 5.2.2 describes the background of distributed computing and our design of the process with ROLDA. Section 5.3 presents the results of our experimental analysis and Section 5.4 concludes with a discussion and areas for future work.

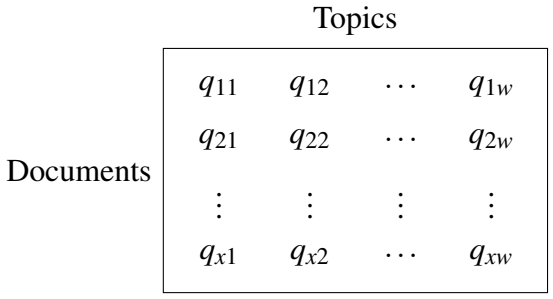
5.1 Methodology

By incorporating the relationships between words and ontology concepts, our improved topic model addresses a significant limitation of OLDA, namely its inability to consider the hidden semantic meanings within the context. We introduce a third dimension to the topic-modelling process using subject-object nouns and concepts relationships extracted from sentences within each document. These cause-effect relationships between labels are able to capture the hidden semantic meanings. For this reason, our technique can be considered an *Relation Incorporated* variant of OLDA, which we abbreviate to ROLDA.



Each document if a collection of relationships between subject nouns and object nouns.

Fig. 5.1 Documents/Subject nouns/Object nouns Δ Matrix Schematic



Each document of distribution of topics.

Fig. 5.2 Documents/Topics Θ Matrix Schematic

$$\Delta = \Theta \times \Sigma \times \Gamma \tag{5.1}$$

As for OLDA, ROLDA's aim is also to generate a documents/topics matrix Θ giving the probability q_{xy} of each document D_x being about a certain topic T_y. The incorporation of the relationships between words and concepts is accomplished by introducing a third dimension in the matrices as follows. We first pre-process the documents $\mathscr{D}$ employing standard open source NLP tools (OpenNLP) for sentence segmentation, part-of-speech (POS) tagging and parsing. By utilising algorithms described in Chapter 4, we extract a set $\mathscr{RN}$ of all cause-effect relations in the documents and a set $\mathscr{N}$ of all nouns in the documents. Each relation $RN \in \mathscr{RN}$ consists of a subject noun $NS \in \mathscr{N}$ and a object noun $NO \in \mathscr{N}$.

As before, we construct the three dimension matrix Δ of binary values, where each cell d_{ijk} is given the value 1 if the document $D_i \in \mathscr{D}$ contains the relation $RN_{jk} \in \mathscr{R}\mathscr{N}$ (i.e. the document D_i contains the the cause-effect relation between the subject noun NS_j and NO_k) or 0, otherwise (this process is described in more detail in Section 5.1.1). Using ConceptNet or DBpedia, we then construct the set of all concepts $\mathscr{C}$ that are associated with a noun $N \in \mathscr{N}$. Each subject (or object) nouns $NS \in \mathscr{N}$ (or $NO \in \mathscr{N}$) can be associated with a subject (object) concept $CS \in \mathscr{C}$ (or $CO \in \mathscr{C}$). Next, we construct a set of relations $\mathscr{R}\mathscr{C}$ of all cause-effect relations between concepts. Each relation $RC \in \mathscr{R}\mathscr{C}$ consists of a subject concept $CS \in \mathscr{C}$ and an object concept $CO \in \mathscr{C}$. It is now possible to construct the four dimensional matrix Γ of binary values, where each cell s_{mneo} is given value 1 only if the dataset contains the relation $RC_{em} \in \mathscr{R}\mathscr{C}$, i.e. the following three conditions are satisfied (this process is described in more detail in Section 5.1.2):

- the subject noun $NS_n \in \mathscr{N}$ can be described by the subject concept $CS_e \in \mathscr{C}$,
- object noun $NO_o \in \mathscr{N}$ can be described by the object concept $CO_m \in \mathscr{C}$,
- the dataset contains the relation $RN_{no} \in \mathscr{R}\mathscr{N}$ between subject noun $NS_n \in \mathscr{N}$ and object noun $NO_o \in \mathscr{N}$

The matrix Σ giving the probabilities r_{abc} of each topic T_a being described by each relation between concepts $RC_{bc} \in \mathscr{R}\mathscr{C}$ is constructed using a logistic regression technique. Finally, Θ is computed by a supervised learning method using Δ, Σ and Γ (the computation of Σ and Θ are described in Section 5.1.3). This overall matrix equation is shown in Equation 5.1. Figure 5.1, 5.2, 5.3 and 5.4 shows the matrix schema with their dimensions.

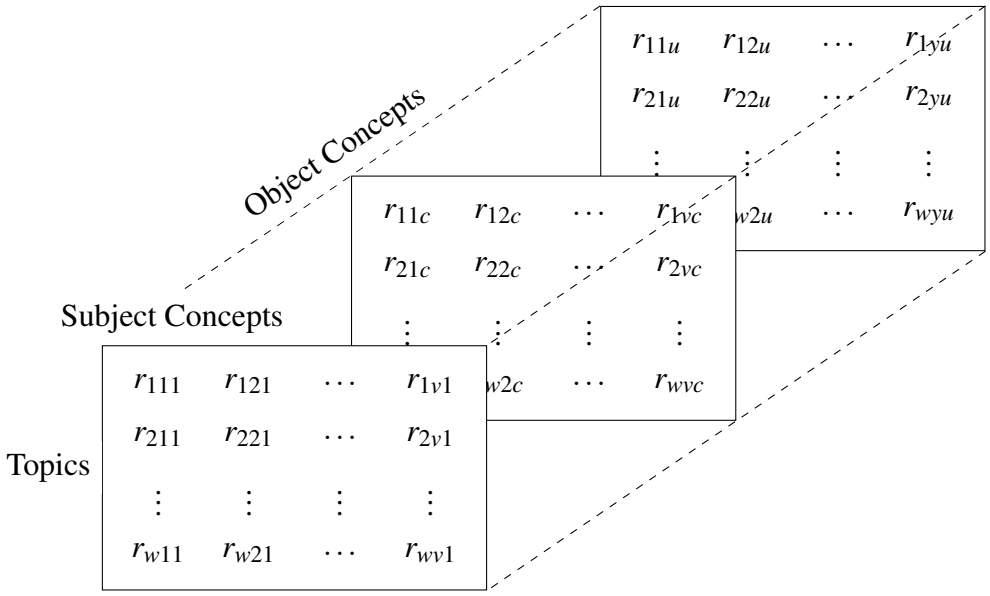
5.1.1 Generating the Documents/subject nouns/object nouns Matrix Δ

The document/subject noun/object noun matrix Δ is a three dimensional matrix consisting of all the relations extracted from documents. The construction of this matrix takes two steps:

- obtaining the set $\mathscr{R}\mathscr{N}$ of all cause-effect relations embedded in sentences and the set $\mathscr{N}$ of all nouns embedded in relations from each document $D \in \mathscr{D}$,
- assign 1 or 0 to each cell based on the existence of relations for each document.

The first step is done by the approaches described in Chapter 4 [Hao et al., 2017]. Each document $D \in \mathscr{D}$ is processed following the extraction process in Figure 4.1. Firstly, standard NLP techniques from OpenNLP [Kottmann et al., 2011] [1] are used to pre-process the text data,

[1] https://opennlp.apache.org



Each topic is a distribution of relationships between subject concepts and object concepts.

Fig. 5.3 Topics/Subject concepts/Object concepts Σ Matrix Schematic

such as tokenisation, lemmatisation, part-of-speech (POS) tagging and parsing. Sentences in each document are re-organised into Penn Treebank style with the corresponding POS tagging. Then NER techniques are applied to extract named entities (nouns). Different NER tools and corpus can be used for documents from different domains: LingPipe, MetaMap and Abner can be used for biomedical domains to recognise DNA, RNA, cell type, body part, etc., Standford NLP offers a NER tool that can be used for general domains to recognise Location, Person, Organisation, etc. [Finkel et al., 2005]. Then following Algorithm 3, 4, 5 and 6, relations embedded in single sentences are extracted. Relations embedded in noun-preposition phrases can also be extracted following rules define in Section 4.4.2. Finally, polarity adjustment is also utilised to further fine-tune the extracted relations and construct the set $\mathscr{RN}$ of all cause-effect relations. Each relation $RN \in \mathscr{RN}$ is in the format of *"Subject Noun NS | Relation Connection | Object Noun NO"*. The set $\mathscr{N}$ consists of all the subject nouns *NS* and object nouns *NO*. For example, four relations $RN(Example\ 5.4)$ can be extracted from Example 5.4, as shown in Table 5.1. Four subject nouns can be extracted from these relations: $NS(Example\ 5.4) = \{$*Zoroaster, Animal, Zoroastrians, Ahura Mazda*$\}$. And four object nouns can be extracted from these relations: $NO(Example\ 5.4) = \{$*Culture, Gods, Ahura Mazda, Lord of Wisdom and Light*$\}$. In total, seven nouns can be extracted from these relations: $N(Example\ 5.4) = \{$*Zoroaster, Animal, Zoroastrians, Ahura Mazda, Culture, Gods, Lord of Wisdom and Light*$\}$.

Each subject concept is associated with subject nouns. Each object concept is associated with object nouns.

Fig. 5.4 Subject Concepts/Object Concepts/Subject Nouns/Object Nouns Matrix Γ Matrix Schematic for a Object Nouns $NO_o \in \mathcal{N}$

Example 5.4. *"Their prophet, Zoroaster, seeking to make sense of a culture in which animal sacrifice to multiple gods was common, preached that there was only one god, a good one. Zoroastrians call their god Ahura Mazda, which translates as Lord of Wisdom and Light."*

Table 5.1 Extracted relations between nouns $RN(Example\ 5.4)$

Subject noun (NS)	Relation Connection	Object noun (NO)
Zoroaster	Seeking to make sense of	Culture
Animal	Sacrifice to	Gods
Zoroastrians	Call	Ahura Mazda
Ahura Mazda	Translates	Lord of Wisdom and Light

We then set $\mathscr{RN} = \bigcup_{NS\in\mathscr{N},NO\in\mathscr{N}} RN(NS,NO)$; assume a fixed ordering of subject nouns $[NS_0,NS_1,\ldots,NS_j,\ldots,NS_y]$ (for $NS_j \in \mathscr{N}$) and a fixed ordering of object nouns $[NO_0,NO_1,\ldots,NO_k,\ldots,NO_z]$ (for $NO_k \in \mathscr{N}$); and then construct each cell d_{ijk} in the matrix Δ for document $D_i \in \mathscr{D}$ follows Equation 5.2.

$$d_{ijk} = \begin{cases} 1 & \text{if } D_i \text{ contains } RN_{jk} \in \mathscr{RN} \\ 0 & \text{if } D_i \text{ does not contain } RN_{jk} \in \mathscr{RN} \end{cases} \tag{5.2}$$

5.1.2 Generating the Subject Concepts/Object Concepts/Subject Nouns/Object Nouns Matrix Γ

The Subject Concepts/Object Concepts/Subject Nouns/Object Nouns Matrix Γ is a four dimensional matrix consisting of all the relations between concepts that are associated with their corresponding nouns. The construction of this matrix takes two steps:

- obtaining the set $\mathscr{RC}$ of all relations between concepts and the set $\mathscr{C}$ of all concepts associated with the set $\mathscr{N}$,
- assign 1 or 0 to each cell based on the existence of concepts relations.

In order to associate nouns to their corresponding concepts, *IsA* and *RelateTo* properties from ConceptNet and *type* properties from DBpedia are extracted automatically using the Internet information extraction techniques described in Section 3.2.1. As for OLDA, we query ConceptNet or DBpedia to obtain the ontological concepts of each subject/object noun $NS/NO \in \mathscr{N}$ and then construct the set of concepts $CS/CO(NS/NO)$ associated with the subject/object noun NS/NO. The same data cleansing approaches are also employed here. For example, the subject noun "Ahura Mazda" has eight different Type properties

in DBpedia: *Thing; Abstraction100002137; Cognition100023271; Concept105835747; Content105809192; Idea105833840; PsychologicalFeature100023100; WikicatConceptionsOfGod*. After data cleansing, we can obtain the set of concepts $CS(AhuraMazda)$ =*{Thing, Abstraction, Cognition, Concept, Content, Idea, Psycological Feature, Conceptions of God}*. The object noun "Ahura Mazda" has the same set of concepts $CO(AhuraMazda)$ as the subject noun "Ahura Mazda". For a noun that does not have any required ontological concepts in ConceptNet or DBpedia, its set of concepts only contains the word itself. For example, $CO(LordofWisdomandLight)$ =*{Lord of Wisdom and Light }*.

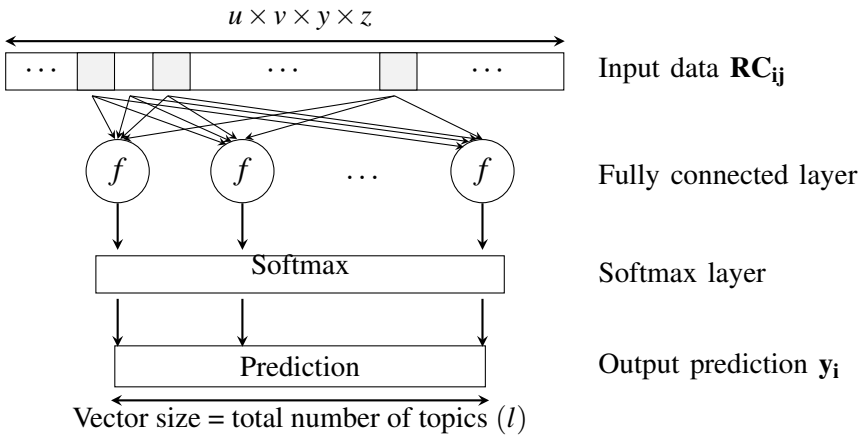
The relations between concepts are then constructed. Each relation $RC \in \mathscr{RC}$ is in the format of *"Subject Concept CS | Relation connection | Object Noun CO"*. Each subject concept of the subject noun $CS(NS)_e \in CS(NS)$ can be associated with every object concept of the object noun $CS(NO)_m \in CS(NO)$. For example, eight *RC* can be derived from the relation $RN(AhuraMazda|Call|LordofWisdomandLight)$ as shown in Table 5.2. The set $\mathscr{RC}$ consist of all concepts relations, and the set $\mathscr{C}$ consists of all the subject concepts *CN* and object concepts *CO*.

Table 5.2 Relations between concepts $RC(AhuraMazda, LordofWisdomandLight)$

Subject concept (CS)	Relation Connection	Object concept (CO)
Thing	Translates	Lord of Wisdom and Light
Abstraction	Translates	Lord of Wisdom and Light
Cognition	Translates	Lord of Wisdom and Light
Concept	Translates	Lord of Wisdom and Light
Content	Translates	Lord of Wisdom and Light
Idea	Translates	Lord of Wisdom and Light
Psycological Feature	Translates	Lord of Wisdom and Light
Conceptions of God	Translates	Lord of Wisdom and Light

Next, we set $\mathscr{RC} = \bigcup_{CS(NS)\in\mathscr{C}, CO(NO)\in\mathscr{C}} RC(NS,NO)$; assuming a fixed ordering of subject concepts $[CS_0, CS_1, \ldots, CS_e, \ldots, CS_v]$ (for $CS_e \in \mathscr{C}$) and a fixed ordering of object nouns $[CO_0, CO_1, \ldots, CO_m, \ldots, CO_u]$ (for $CO_m \in \mathscr{C}$); and then construct each cell s_{mneo} in the matrix Γ follows Equation 5.3.

$$s_{mneo} = \begin{cases} 1 & \text{if } RC_{em} \in \mathscr{RC} \\ 0 & \text{if } RC_{em} \notin \mathscr{RC} \end{cases} \tag{5.3}$$



The input is a vector of relations between concepts $\mathbf{RC_{ij}}$ for word W_i. The other layers are same as Fig 3.3.

Fig. 5.5 *Structure of logistic regression model*

5.1.3 Generating the Matrices Θ and Σ

The documents/topics matrix Θ and the topics/subject concepts/object concepts matrix Σ are generated iteratively using the input matrix Γ with a logistic regression model. As in OLDA, the model uses the linear weighted combination of inputs from Γ and generates the predicted probabilities of each relation between subject/object concepts relating to each topic (i.e., the matrix Σ) [Menard, 2002, Walker and Duncan, 1967a]. A schematic diagram of the model is shown in Fig. 5.5. For each relation $RN_{ij} \in \mathscr{RN}$ between subject noun NS_i and object noun NO_j, the corresponding columns of subject concepts $CS(NS_i)$ and object concepts $CO(NO_j)$ in the subject concepts/object concepts/subject nouns/object nouns matrix Γ is used as an input data vector $\mathbf{RC_{ij}}$ (Equation 5.4).

$$\mathbf{RC_{ij}}_k = \begin{cases} 1 & \text{if } RC(NS_i,NO_j)_k \in RC(NS_i,NO_j) \\ 0 & \text{if } RC(NS_i,NO_j)_k \notin RC(NS_i,NO_j) \end{cases} \tag{5.4}$$

The size of the input vector is the total number of object concepts $u\times$ total number of subject concepts $v\times$ total number of object nouns $z\times$ total number of subject nouns y. The output vector $\mathbf{y_{ij}}$ is the predicted probability of each relation between subject/object concepts being associated with a topic, as described next. A fully connected layer takes the vector $\mathbf{RC_{ij}}$ and generates the evidence vector $\mathbf{z_{ij}}$ using Equation 5.5 and a weight matrix $\mathbf{W}_t$ and bias vector b_t. The initial values $\mathbf{W}_0$ and b_0 are randomly given.

$$\mathbf{z_{ij}} = f(\mathbf{RC_{ij}}) = \mathbf{WRC_{ij}} + b \tag{5.5}$$

Each element $\hat{r}_a$ in the evidence vector $\mathbf{z_{ij}}$ is then normalised in the softmax layer to finally generate the vector $\mathbf{y_{ij}}$ according to Equation 5.6 (this means that the values within $\mathbf{y_{ij}}$ add up to 1). Each element r_{abc} in the output vector $\mathbf{y_{ij}}$ is the predicted probability of each relation RC_{bc} between subject concept CS_b and object concept CO_c being associated with a topic T_a. This whole process is repeated for all relations between subject nouns and object nouns ($i \in (1,2,\ldots,y)$ and $j \in (1,2,\ldots,z)$), resulting in the matrix Σ_t. Finally, the matrix Θ_t can be computed using the matrix schematic shown in Fig. 3.2.

$$\mathbf{y_{ij}} = softmax(\mathbf{z_{ij}}) = \frac{e^{\hat{r}_a}}{\sum_{a=1}^{w} e^{\hat{r}_a}} \tag{5.6}$$

The iteration process works the same as OLDA. The initial matrices Σ_0 and Θ_0 are obtained using random values for the weight matrix $\mathbf{W}_0$ and the bias vector b_0. For each subsequent iteration $t+1$, we then measure the Euclidean distance between the predicted classification Θ_t and the true classification Θ_s (recall Θ_s is manually done). Using the Stochastic Gradient Descent technique we obtain new values for $\mathbf{W}_{t+1}$ and the vector b_{t+1} [Bottou, 1998, Kiefer et al., 1952] that minimise this distance. We then calculate Σ_{t+1} and Θ_{t+1} as before using $\mathbf{W}_{t+1}$ and b_{t+1}. This process continues until the distance between the predicated classification Θ_j computed in an iteration j, and the true classification Θ_s goes below the desired threshold. The output of this process is the documents/topics matrix Θ, the topics/subject concepts/object concepts matrix Σ, and the optimised weight matrix $\mathbf{W}$ and bias vector b.

5.2 Topic Classification with Distributed Computing

In order to further speed up the training process of the topic model, a distributed computing process is introduced. In this section, we describe a distributed computing process that can handle the computation of large scale matrices and accelerate the computing process. We first describe the background of distributed computing and cloud computing in Section 5.2.1. Then in Section 5.2.2 ROLDA is combined with a distributed computing process.

5.2.1 Background

Distributed computing studies distributed systems, whose components are located on different networked processors, each with their own local memory to communicate and coordinate

their actions by passing messages to one another [Tanenbaum and Van Steen, 2007]. These components interact with one other aiming to achieve a common goal. Distributed computing can achieve a large and complex computation through leveraging computing resources from different locations connected by networks [Wu and Buyya, 2015]. Fig 5.6 shows a typical distributed system structure. Distributed systems have five advantages compared to traditional single computing, vertical scaling computing and parallel computing:

- Scalability and Modular Growth: Distributed systems can scale horizontally as they work across different machines. Traditional computing systems rely on upgrading the hardware to handle increasing workload whereas distributed systems can simply add another machine. When the demand is high, a system can run each machine to the full capacity; when the workload is low, it can take machines offine [Schmidt et al., 1999].

- Fault tolerance and redundancy: The distributed system can tolerate failures in individual components [Ghosh, 2014]. It can stay the same reliable even if one or more nodes (processors) stop working, and the performance demand on the remaining nodes would go up.

- Low Latency: Distributed systems prioritise node based on their distance to users geographical locations, resulting in low latency and better performance [Ghosh, 2014].

- Cost Effectiveness: The initial cost of distributed systems is higher than traditional vertical scaling systems, but after a certain point, they are more about economies of scale. A distributed system consisting of many mini computers can be more cost-effective than a mainframe machine. [Peleg, 2000]

- Efficiency: Distributed systems break complex problems/data into smaller pieces and have multiple mini computers working on them in parallel, which can help reduce the time needed to compute those problems [Ghosh, 2014].

Cloud computing is a computer system resource for data storage and computing power without direct active management by the user. Large clouds often have functions distributed over multiple processors from central servers. Cloud computing can achieve coherence and economies of scale by sharing resources. Cloud computing has several advantages compared to computing on local machines[Mell et al., 2011]:

- Agility: The cloud allows innovating faster because resources can be mainly used on developing applications rather than managing infrastructure and data centres[Lin, 2008, Bruneo et al., 2013].

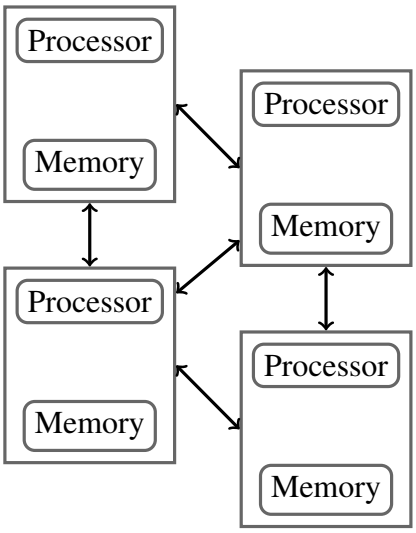


Fig. 5.6 A distributed computing system

- Elasticity: Cloud computing enables the provision of the number of resources required to instantly scale up or down [Mao and Humphrey, 2012, Nouri et al., 2019, Shawky and Ali, 2012].
- Cost savings: The cloud allows trading capital expense (data centres, physical servers, etc.) for variable operational expenditure. Pricing is based on actual usage. As well, less in-house IT skills are required for the implementation of projects that use cloud computing [Knorr and Gruman, 2008, Subramanian, 2009].

Nowadays, many public cloud services are available and open for public use. Generally, public cloud service providers like Amazon Web Services (AWS) [Varia et al., 2014], IBM [Iannucci et al., 2013], Oracle [Saygili, 2017], Microsoft [Copeland et al., 2015], Google [Krishnan and Gonzalez, 2015], and Alibaba [Ren et al., 2017] own and operate the infrastructure at their data centre and users normally get access via the Internet. Different cloud services provide different components for object storage, message queuing and computing platform. In particular, we present three cloud services provided by AWS.

- Amazon Simple Storage Service (Amazon S3): provides object storage through a web service interface [Varia et al., 2014, Huang and Wu, 2017]. Amazon S3 can be used to store a variety type of objects which allows storage for Internet applications, backup and recovery, disaster recovery, data archives, data lakes for analytic, and hybrid cloud storage [Andreozzi et al., 2008].
- Amazon Simple Queue Service (Amazon SQS): is a distributed message queuing service. It sends messages via web service applications for communications over the

Internet. SQS can provide a highly scalable hosted message queue that addresses issues arising from the common producer-consumer problem or connectivity between producer and consumer [Robinson, 2008].

- Amazon Web Services Lambda (AWS Lambda): is an event-driven, serverless computing platform that runs codes in response to events and automatically modifies the computing resources required [Handy, 2014]. The purpose of Lambda is to simplify building smaller, on-demand applications that are responsive to events and new information. AWS aims to start a Lambda instance within milliseconds of an event [Hendrickson et al., 2016].

In Section 5.2.2, ROLDA is combined with a distributed computing process. For experiment purpose, we use these AWS cloud services in our distributed computing process.

5.2.2 ROLDA with Distributed Computing

The construction time and computing resources of building a topic model with ROLDA are massive due to the 3D and 4D matrices calculation process. Therefore, we implemented a distributed cloud computing process to accelerate the computation process.

Generating the Documents/subject nouns/object nouns Matrix Δ

Each document $D \in \mathscr{D}$ is stored in an object storage bucket in the cloud service (in our case, Amazon S3 bucket) instead of a local computer. Then, a distributed message queuing service (in our case, Amazon SQS service) is created using a First-In-First-Out queue. Each document in the input generates a message by the queuing service to communicate between the storage and the computing platform (in our case, Amazon Lambda). When receiving a message, the computing platform takes in the document D, starts to run the codes to extract relations (following Figure 4.1) and outputs relations embedded in the document D. The outputs are stored in a new storage bucket in the cloud service. The output subject nouns of relations are stored in their corresponding document folders, and the output object nouns of relations are stored in their corresponding subject nouns folders within their corresponding document folders. Then, each cell in the matrix Δ can be assigned with 1 or 0 based on the existence of each file stored in the storage bucket. Once a processor finishes the process of one document, the memory is released and ready to accept the next message from the queuing service. Figure 5.7 shows an example of this distributed process.

With this distributed computing process, all documents can be processed at the same time in the cloud. In average, the construction time for generating Δ with distributed cloud system

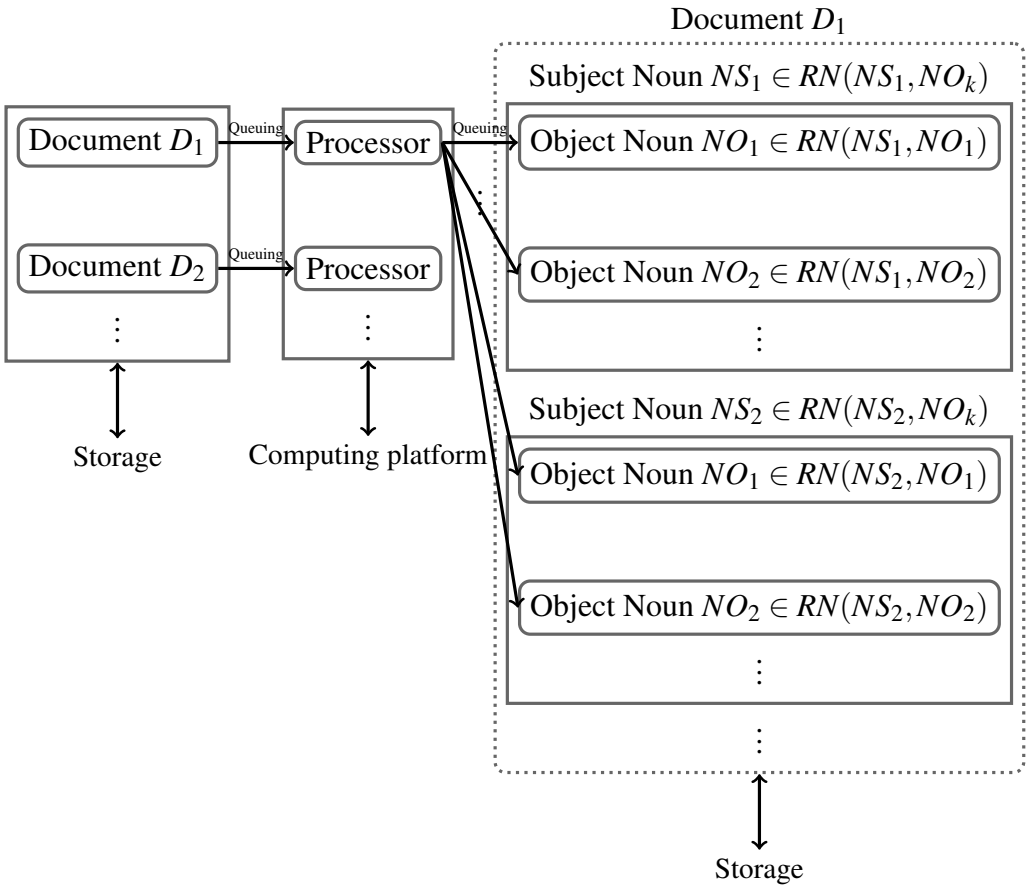


Fig. 5.7 Distributed computing process for Δ

can be reduced to only 5% of a traditional local computing system. The detail information is shown in Section 5.3.

Generating the Subject Concepts/Object Concepts/Subject Nouns/Object Nouns Matrix Γ

For traditional local computing, the construction of the matrix Γ takes the same two steps:

- Obtaining the set $\mathscr{RN}$ of all relations;
- Assign 1 or 0 to each cell. However, for a distributed cloud computing.

These two steps can be done by one processor in a cloud computing platform.

The output data from the previous stage are the input of this stage: each object noun $NO_k \in \mathscr{N}$ are stored under the folder of their related subject noun $NS_j \in \mathscr{N}$ and each subject noun folder are stored under the folder of their corresponding document $D_i \in \mathscr{D}$. All the documents are stored in a storage bucket in the cloud instead of in a local computing machine.

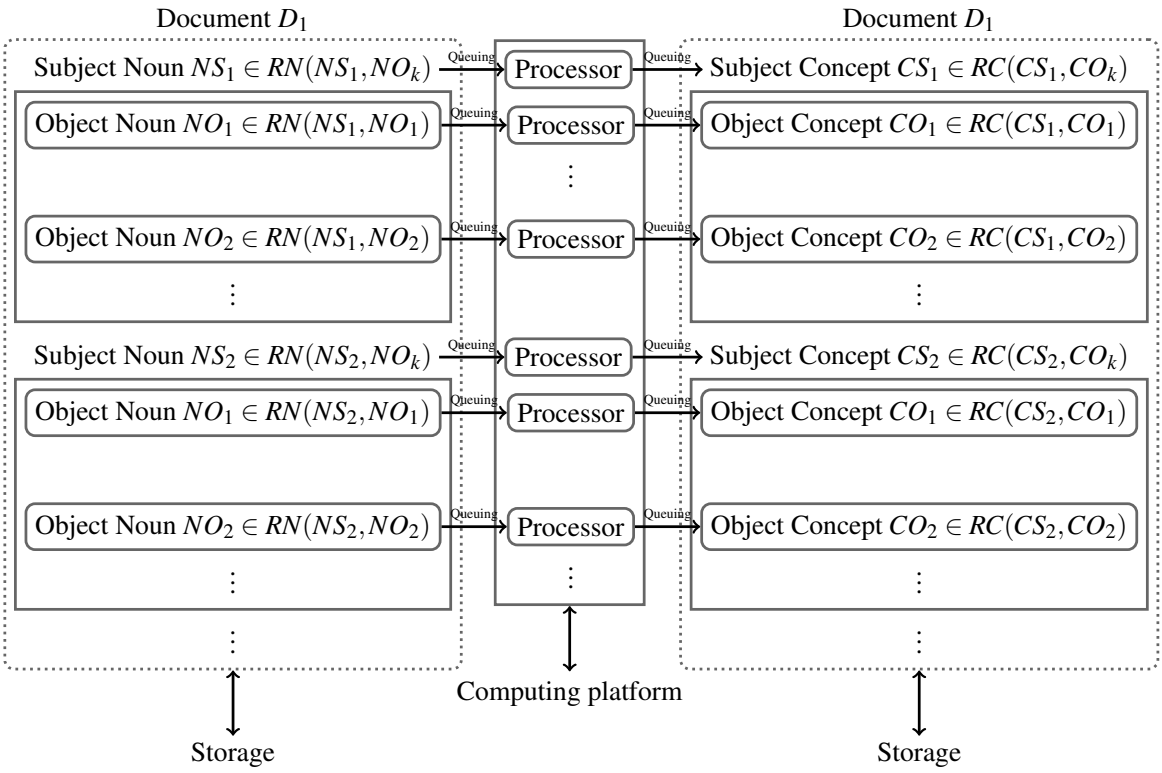


Fig. 5.8 Distributed computing process for Γ

Then a distributed message queuing service is created using First-In-First-Out queue. Each object noun in a document D_i generates a message by the queuing service to communicate between the storage and the computing platform. After all object nouns in the same subject noun NS_j folder have been associated with their object concepts from the ontology, the subject noun generates a message by the queuing service to inform the computing platform running the code. Then the object nouns in the next subject noun NS_{j+1} folder starts to generate messages by the queuing service to run the code in the computing platform. After all subject nouns in document D_i have been associated with their subject concepts from the ontology, the object nouns and subject nouns in the next document D_i generate messages by the queuing service to run the code in the computing platform. Each cell in the matrix Γ can be assigned with 1 or 0 based on the existence of each file stored in the storage bucket. Every time when a processor in the computing platform finishes one process, the memory is released and ready to receive the next message sent from the queuing service. Figure 5.8 shows an example of this distributed process.

With this distributed process, all nouns can be associated with their corresponding concepts at the same time in the cloud. In average, the construction time for generating Γ

with distributed cloud system can be reduced to only 5% of a traditional local computing system. The detail information is shown in Section 5.3.

Generating the Matrices Θ and Σ

The Document/topics matrix Θ and the topics/subject concepts/object concepts matrix Σ are generated iteratively using the input matrix Γ in a logistic regression model. The training process of the logistic regression model is set up in a cloud GPU. The input is stored in two storage buckets, one contains the subject nouns and object nouns for each document, and the other contains the subject concepts and object concepts for each document. For each iteration, the queuing service generates $u \times v \times y \times z$ messages to generate the input vector $\mathbf{RC_{ij}}$ from the storage buckets and start training the logistic regression model. The training process is the same as in the local computing machine. This process continues until the distance between the predicated classification Θ_j computed in an iteration j, and the true classification Θ_s goes below the desired threshold. The output of this process is the documents/topics matrix Θ, the topics/subject concepts/object concepts matrix Σ, and the optimised weight matrix **W** and bias vector b.

5.3 Experiment Analysis

As for OLDA, ROLDA can also be used with or without a self-training stage. When self-training is employed, the resulting topic model can be referred to as ST-ROLDA instead. Those two different self-training procedures ST can be subscripted with H (to indicate the use of the ad hoc training procedure) or P (to indicate the use of Pavlinek et. al.'s). Where the distinction is irrelevant, we avoid the subscript. With all this in mind, we conducted comprehensive benchmarking to evaluate the performance of our ROLDA, OLDA and variants against a number of other semi-supervised methods using the same four datasets.

More specifically, we compared the classification accuracy of ROLDA with the performance of OLDA. Furthermore, we considered the two self-training techniques (ST_H and ST_P) described in Section 3.3 for OLDA and ROLDA, resulting in a total of four variations of the semi-supervised methods. Both ontologies from ConceptNet and DBpedia are used with OLDA and ROLDA for experiments. The results of our experiments are given in Tables 5.4. Table 5.5 shows the time to construct a topic model using 20 Newsgroups dataset.

5.3.1 Experimental Setup

To perform a comparative analysis, four fully classified training datasets (presented below) were used. The experimental setup in this phase is similar to Section 3.4.1. Each dataset was split into a training dataset (50%-70%) and a testing dataset (30%-50%). For each round of supervised experiments, all the training datasets were used to construct the topic model and the supervised classification method SVM. For each round of the semi-supervised experiments, 10% of the training datasets denotes the initial pre-classified dataset D_s, and the remaining training data form the unclassified datasets D_u. The self-training topic model was then used to prepare the final classified datasets D_{ss}, which were trained with a supervised classification method SVM. The trained classifiers were finally evaluated on the testing datasets.

Both self-training algorithms were implemented in the Java programming language using WEKA [Hall et al., 2009], which is an open-source machine learning environment. LibSVM implementation was used to train an SVM classifier with a linear kernel on the final classified dataset [Chang and Lin, 2011]. All experiments were performed on a PC with an i7 processor, an NVIDIA GeForce GPU GTX 970M graphics card, and 16GB RAM. All experiments with distributed computing were set up on the Amazon cloud. In our experiments, the cloud storage bucket is Amazon S3 bucket, the distributed message queuing service is Amazon SQS, and the computing platform is AWS Lambda.

5.3.2 Datasets Used in the Analysis

The same four datasets were used in this analysis: 20 Newsgroups dataset, the Reuters R8 and R52 datasets and the WebKB dataset. The same pre-processing steps explained in Section 3.4.2 were performed on them. The same number of words were extracted from each dataset. After named entity recognition and relation extraction, subject nouns and object nouns were extracted to construct the documents/subject nouns/object nouns matrix Δ. Then concepts were extracted from either ConceptNet or DBpedia and associated with these nouns to construct the subject concepts/object concepts/subject nouns/object nouns matrix Γ. Table 5.3 shows the size of each matrix for each dataset. The same portion of data from each dataset explained in Section 3.4.2 were used for evaluation experiments.

5.3.3 Experimental Results

For ontologies from ConceptNet and DBpedia, we conducted two rounds of experiments with each of the four datasets. For the supervised approaches, we skipped the self-training phase

Table 5.3 The size of each matrix for each dataset

Dataset	documents/subject nouns/object nouns matrix Δ	subject concepts/object concepts/subject nouns/object nouns matrix Γ	
		ConceptNet	DBpedia
20 Newsgroups	$18,846 \times 132,385 \times 137,824$	$12,311 \times 13,198 \times 132,385 \times 137,824$	$12,450 \times 13,215 \times 132,385 \times 137,824$
Reuters R8	$7674 \times 7522 \times 7609$	$4989 \times 5114 \times 7522 \times 7609$	$5011 \times 5209 \times 7522 \times 7609$
Reuters R52	$9100 \times 7397 \times 7388$	$5963 \times 5989 \times 7397 \times 7388$	$6011 \times 6076 \times 7397 \times 7388$
WebKB	$4199 \times 6691 \times 6421$	$4809 \times 4924 \times 6691 \times 6421$	$5009 \times 4899 \times 6691 \times 6421$

and used the proportions of data described in Section 5.3.2. In each round of the experiments, we performed 10 repetitions in the training and selected the data for training using stratified random sampling for each topic category, so that each topic had equal representation in the training set.

Table 5.4 summarises the classification accuracy results of EM-NB, OLDA and ROLDA when using supervised training procedure and either of the two self-training procedures ST_H and ST_P. Table 5.5 summarises the topic model's construction times for each technique for the 20Newsgroup dataset.

In what follows, we discuss the results using each self-training procedure in more detail.

Supervised training

As we mentioned, we can skip the self-training phase in our method resulting in a fully supervised classification engine that we simply refer to as ROLDA. We compared ROLDA's accuracy with that of the supervised OLDA and EM-NB approach [Nigam et al., 2000]. Our results show that EM-NB performed worst in all datasets, especially in Reuters R52. These results are expected as EM-NB tends to perform poorly when classifying documents into a larger set of topic categories. As shown in Table 5.4, ROLDA outperforming OLDA by quite a considerable margin regardless of ontologies (e.g., with ConceptNet ontology, 72.54% against 67.11% in the Reuters R52 dataset; with DBpedia ontology, 73.11% against 68.02% in the same dataset). With either ontology, ROLDA achieves around the same accuracy, which means ROLDA can be applied in different domains with different ontologies.

As shown in Table 5.5, the construction of the topic model for the 20 Newsgroups dataset using ROLDA only took about 3 days to complete while it took 16 days for EM-NB. Even though the matrix computation process of ROLDA is more completed than OLDA, the fact that the construction time of ROLDA only took $3/4$ of the time for OLDA shows the efficiency of the introduction of distributed cloud computing. The construction times of EM-NB and OLDA are about half of the time for that on a local machine (as shown in Table 3.2) further confirm the efficiency of the introduction of distributed cloud computing.

Table 5.4 Classification accuracy results (CI=95%)

Dataset			20Newsgroup	Reuters R8	Reuters R52	WebKB
Supervised	EM-NB		53.12%	34.12%	26.90%	55.56%
	OLDA	ConceptNet	89.86%	87.47%	67.11%	85.21%
		DBpedia	90.22%	87.90%	68.02%	85.87%
	ROLDA	ConceptNet	**92.79%**	**90.98%**	**72.54%**	**88.65%**
		DBpedia	**93.44%**	**91.22%**	**73.11%**	**89.21%**
Semi-supervised ST_H	OLDA	ConceptNet	71.76%	75.89%	56.02%	74.75%
		DBpedia	72.12%	77.32%	56.14%	74.90%
	ROLDA	ConceptNet	**75.01%**	**77.57%**	**62.32%**	**76.42%**
		DBpedia	**75.31%**	**78.25%**	**62.90%**	**77.11%**
Semi-supervised ST_P	OLDA	ConceptNet	77.67%	82.86%	64.08%	80.79%
		DBpedia	78.01%	83.11%	64.25%	81.89%
	ROLDA	ConceptNet	**80.43%**	**85.08%**	**70.09%**	**82.35%**
		DBpedia	**81.55%**	**85.90%**	**70.23%**	**83.55%**

Table 5.5 Time to construct the 20 Newsgroups topic model

Technique		Construction time (days)	
		Local machine	Distributed cloud computing
Supervised	EM-NB	30	16
	OLDA	8	4
	ROLDA	NA	3
Semi-supervised ST_H	OLDA	2	2
	ROLDA	NA	1
Semi-supervised ST_P	OLDA	5	4
	ROLDA	NA	2

Self-Training Using the Simplified Approach (ST_H)

As we mentioned in Section 3.3.1, the training procedure stops when the distance between the predicted and actual classification drops below a certain threshold. In our experiments for ROLDA, this distance drops dramatically in the first $2,500,000$ iterations, decreasing further but at a reduced rate in later iterations. The distance remained fairly stable after $20,000,000$ iterations dropping to values close to 0.44. For that reason, we stop iterating when the distance goes below 0.45. As for OLDA, we stop iterating when the distance goes below 0.55 after around $20,000,000$ iterations.

As shown in Table 5.4, ROLDA outperforms OLDA by quite a considerable margin regardless of ontologies (e.g., with ConceptNet ontology, 62.32% against 56.02% in the Reuters R52 dataset; with DBpedia ontology, 62.90% against 56.14% in the same dataset). With either ontology, ROLDA achieves around the same accuracy, which means ROLDA can be applied in different domains with different ontologies. As shown in Table 3.2, the construction of the topic model for the 20 Newsgroups dataset using the training procedure ST_H for ROLDA took one day to complete, which is the same for OLDA regardless of the more complicated computing process. Furthermore, the construction time for OLDA with ST_H on distributed computing process was reduced to one day, which is half of that on local machines and 20% of LDA (see Table 3.2).

Self-Training Using Pavlinek et al.'s Approach (ST_P)

ROLDA's construction of the topic model for the 20Newsgroup dataset using the training procedure ST_P took about three and a half days, whilst OLDA's took three days. That is, ROLDA's construction took around the same time as OLDA regardless of a more complicated computing process. This is also 60% of OLDA on local machines and 30% of LDA (see Table 3.2).

Table 5.6 Topic classification results of state-of-the-art work on 20Newsgroup dataset

Model		Accuracy
Word embedding based	LFLDA	80.94%
	<word, POS> embedding model	83.05%
Knowledge based	ST_P-OntoLDA	72.41%
	ST_P-ROLDA	81.55%

In terms of accuracy, the training procedure of ST_P performed better than when using ST_H in all techniques and datasets. The best combination was ST_P and ROLDA, which outperformed ST_P and OLDA by quite a considerable margin regardless of ontologies (e.g., with ConceptNet, 70.09% against 64.08% in the Reuters R52 dataset; with DBpedia, 70.23% against 64.25% in the same dataset). With either ontology, ROLDA achieves around the same accuracy, which means ROLDA can be applied in different domains with different ontologies.

Table 5.6 compares the classification accuracy results of ST_P-ROLDA against some state-of-the-art work on 20Newsgroups dataset: including both word embedding based approaches (LFLDA [Fu et al., 2016] and <word, POS> embedding model [Liu et al., 2019]) and knowledge-based approaches (ST_P-OntoLDA [Allahyari and Kochut, 2015]). Our proposed ST_P-OLDA increases the accuracy of classification by 0.61% compared to LFLDA and achieves slightly lower accuracy compared to <word, POS> embedding model. Similarly to ST_P-OLDA, our proposed ST_P-ROLDA still benefits when dealing with small corpus as it does not rely on external word embeddings. Comparing against the state-of-the-art knowledge-based approach ST_P-OntoLDA, the introduction of the relation component into the topic model further increases the classification accuracy by 9%.

5.4 Summary

OLDA described in Chapter 3 has certain advantages compared to conventional data-driven approaches. Firstly, OLDA uses the semantical meanings of the words and integrating the fact that individual words may have multiple meanings and that different words may have the same meaning. This enables the ability of OLDA to perform the modelling independently of the particular set of words describing the topics. Secondly, OLDA can be trained with a self-training procedure, which reduces the amount of classified training data for supervised machine learning. For conventional approaches, generating the training data is expensive and time-consuming as it relies on humans to collect, read and manually classify the data in a consistent manner. However, OLDA still has some drawbacks. Firstly, the introduction

of all semantical meanings of words also introduces irrelevant information in the context. OLDA ignores semantical structures in texts, such as the relationships between words and semantical meanings. Secondly, the construction time of the topic model using OLDA (with or without the self-training process) still took quite a long time.

In this chapter we propose a novel approach based on OLDA that uses not only ontological information about the semantical meaning of the words but also integrates relations between these ontological information embedded in documents, allowing topics to be represented more faithfully and independently to the particular set of words used to describe them. This approach, that we called Relation-Ontology-Driven Latent Dirichlet Allocation (ROLDA), can be combined with a self-training phase to produce a semi-supervised method (ST-OLDA), which requires only a small amount of pre-classified training data. We also developed a distributed cloud computing process that can be used for ROLDA and OLDA, which reduced the training time required.

Our experiments, using the four datasets “20 Newsgroups”, “Reuters R8”, “Reuters R52" and “WebKB”, show that the addition of the relationships between concepts into OLDA significantly increases the accuracy of the classification. In addition, when used with distributed cloud computing, this significantly reduces the time required for training.

Our main conclusions can be summarised as follows:

1) The inclusion of the relationship component reduces the self-training time by nearly half using the supervised procedure and two distinct self-training procedures. In particular, it reduces the time needed for training using the self-training procedure proposed by [Pavlinek and Podgorelec, 2017] by nearly half in the 20 Newsgroups dataset.

2) The inclusion of the relationship component also increases the accuracy of the classification regardless of the self-training method employed by between 1 and 6 percentual points (depending on the training method and dataset).

3) The inclusion of the distributed cloud computing process significantly reduces the training time by nearly half using the supervised procedure and two distinct self-training procedures while retaining the high classification accuracy.

Chapter 6

Conclusion and Future Work

This thesis looked for answers on the questions about topic modelling and classification, especially, how one topic is different from others; what makes the documents being grouped into one topic; if semantical meanings of words in a document matter in topic modelling; if the context is important in topic modelling; and what tools and machine learning techniques work best for topic classification.

Automatic classifying documents into topics need computers to look not only at the occurrence of words but also at their meaning in the document. However, conventional topic classification techniques treat topics as a bag of words, ignoring their semantical meanings and context. During this PhD research, three main contributions were proposed:

- Incorporate ontology knowledge with LDA for topic classification to consider semantical meanings of unstructured texts.
- Extract structured relations from unstructured texts using an entity-based algorithm to capture the semantical structures of unstructured texts. As a side contribution of this work, we also created a new dataset from Pubmed for relation extraction task in the biomedical domain.
- Combine the ontology knowledge and the extracted structured relations with LDA for topic classification to consider both the semantical meaning and semantical structures in texts.

The proposed relation extraction algorithm was published in a conference paper entitled “An Entity-Based Algorithm for Multiple-Relation Extraction from Single Sentences” [Hao et al., 2017]. The proposed topic classification incorporating ontology knowledge with LDA was presented in a conference entitled “A self-Training Ontology-Driven Approach for Topic Classification (ST-OLDA)”.

6.1 To incorporate ontology knowledge with LDA for topic classification

In order to consider the semantical meanings of words when constructing a topic model, we proposed a novel topic modelling approach based on LDA that uses ontological information obtained from ConceptNet or DBpedia about the semantical meaning of the words. Chapter 3 explains this approach, that we called *Ontology-Driven Latent Dirichlet Allocation* (OLDA). By associating ontology knowledge, OLDA allows the topics to be defined more generally in terms of ontological concepts rather than words. The proposed ontology-driven topic model improves the topic coherence in comparison to the standard LDA model by integrating ontological concepts with probabilistic topic models into a unified framework. The inclusion of these ontological information also helps to increase the accuracy of the classification and reduce the training and classification times.

In order to further reduce the amount of manually classified data needed and to increase the speed of construction the topic model, we introduced a self-training phase, which can produce a large amount of classified data from the relatively small amount of manually classified data. This variant using the self-training phase is called *Self-Training Ontology-Driven Latent Dirichlet Allocation* (ST-OLDA). Two alternatives in the self-training phase were considered: a relatively ad hoc method employing a logistic regression model and Pavlinek procedure using Gibbs sampling [Pavlinek and Podgorelec, 2017]. The inclusion of this self-training procedure enables OLDA to construct the topic model using only 10% of manually pre-classified data and the remaining 90% of unclassified data.

Two rounds of experiments with each of four datasets, “20 Newsgroups”, “Reuters R8”, “Reuters R52” and “WebKB” were conducted for comparison with OLDA. We re-implemented three existing topic modelling approaches: EM-NB, TF-IDF and LDA. Self-training techniques were employed with the latter three approaches, resulting in six variations in our experiments. In order to experiment with different ontologies, ConceptNet and DBpedia were used for OntoLDA and OLDA. These topic models were then combined with an SVM classifier to perform topic classifications. The experiments results, shown in Table 3.1, Table 3.2 and Table 3.3, confirms that the addition of the semantical component into LDA significantly increases the classification accuracy. When used with self-training, these ontology knowledge reduces the required amount of manually classified training data and also increases the performance of topic classification.

6.2 To extract structured relations from unstructured texts using an entity-based algorithm

In order to capture the semantical structures of unstructured texts, automatic extract relation information embedded in text documents is important. In Chapter 4, our proposed entity-based algorithm for relation extraction is presented. Unlike machine learning approaches and rule-based approaches, the proposed algorithm does not require a large amount of manually annotated training data or domain rules. Conventional rule-based approaches that fail to capture relationships embedded in complex sentence structures as they focus on verbs or relation connection words. Unlike them, the proposed algorithm can identify multiple relationships based on the existence of multiple entities within a single sentence. By utilising standard NLP techniques, the grammatical structure of the sentences are considered to identify and extract relationships embedded within complex structures, including clauses, conjunctions, and noun-preposition phrases. The algorithm also employed a relationship polarity adjustment function so that it takes into account adjectives and adverbs modifying relations' intended meanings.

Five main improvements were employed in our entity-based algorithm:

- It replaced pronouns with their corresponding bio-entities allowing the extraction of relationships that would otherwise be missed. By employing a co-reference resolution component, chains between pronouns and their corresponding nominal words can be formed. This step enables a better accuracy of Named Entity Recognition, which results in better accuracy of relation extraction;

- It added the ability to extract relationships embedded in semantically similar verbs, which are synonyms provided by UMLS, WordNet and VerbNet lists. The extracted relation structures are the same as a conventional verb-based approach, which is a verb-centric tuple *Entity*| *Relation connection* | *Entity*. Except that the relation connection words in our approach are enlarged by the semantical similar corpus;

- It added the ability to extract multiple relationships embedded within certain types of sentence structures. Specifically, three sentence structures were considered: clauses structures, sentence level conjunctive structures and phrase level conjunctive structures. By identifying these structures and processing them into smaller semantic units, multiple verb-centric relations embedded in complex long sentences can be extracted;

- It extracted relationships embedded within noun-preposition phrases. Specifically, three patterns of noun-preposition phrases were considered. By identifying these

structures, relations embedded in noun-preposition phrases can be extracted, which means our algorithm can deal with most of the commonly used sentence structures;

- It determined the relationship polarity and fine-tuned the process by excluding relationships that have not been explicitly asserted in the text. By defining the concept of *relationship polarity*, the polarity score of relation connection words can be computed by using SentiWordNet so that the extracted relation can be associated with three possible polarities: positive, negative or neutral. Only recording those with positive polarities enable a better understanding of the texts.

In order to evaluate the performance of our approach, we analysed the extraction results of algorithms using different combinations of contributions. Table 4.9 shows the complete set of results on two datasets from different domains, which show that our proposed algorithm performances the conventional rule-based algorithms by a large margin (overall precision of 0.19 and recall of 0.297). The experiments using different datasets also confirms that the proposed algorithm can be easily applied in new domains with training corpus. This algorithm enables the machine to better capture the structured information embedded in the unstructured text data.

6.3 To combine the ontology knowledge and the extracted structured relations with LDA for topic classification

By combining the ontology knowledge and the extracted structured relations with the topic model LDA, we proposed the topic modelling approach in Chapter 5 so that both the semantical meaning and semantical structures embedded in the texts can be considered. This approach that we called *Relation Incorporated Ontology-Driven Latent Allocation* (ROLDA), allows topics to be represented more faithfully and independently in terms of relations of ontology concepts rather than the particular set of words. We utilised the relation extraction algorithm proposed in Chapter 4 to automatically extract relations from text documents. The inclusion of the relationship component increases the topic classification accuracy and reduces the required time for topic model construction. As for OLDA, ROLDA can also be combined with a self-training phase to achieve a semi-supervised method (ST-OLDA).

In order to further increase the speed of the training procedure, we employed a computing process of ROLDA and OLDA in a distributed manner. The inclusion of distributed cloud computing significantly reduces the training time by nearly half. This distributed computing process can be used in various topic modelling and classification methods to accelerate the computing process.

Two rounds of experiment with each of four datasets as OLDA were conducted. The experiments results, shown in Table 5.4, Table 5.5 and Table 5.6, confirms that the addition of the relation component into OLDA significantly increase the classification accuracy. When used with self-training, ST-ROLDA reduces the required amount of manually classified training data and also increases the performance of topic classification. When used with the distributed cloud computing process, both ROLDA and OLDA requires nearly half of time for training.

6.4 Application

The relation-ontology driven topic classification has an important significance for the research community at large, such as information retrieval and recommendation systems [Paul and Girju, 2009]. This approach can be very useful to computational linguists that can extract novel ideas of novel topics, generate theoretical models from different linguistics, build sophisticated systems and apply them to Education [Reisenbichler and Reutterer, 2019]. This can be intensified in the future—both on the implementation of topic models and on the level of extending the method itself. For example, this topic model can be shifted from an exploitative method to one that solves hypotheses testing problems by modifying the outputs of the topic model into two subsequent models [Sun et al., 2013].

Sentiment Analysis is probably the most common example of topic classification [Iman et al., 2017, Recalde and Baeza-Yates, 2018]. It is a computational approach aiming to identify opinion, sentiment, and subjectivity in text. Rather than classify documents based on their topics, sentiment classification classifies them with opinions, such as positive or negative or neutral [Ravi and Ravi, 2015]. According to a survey from Hootsuite, around half of Americans have interacted with companies or institutions on at least one of the commonly used social media networks [Kinsky et al., 2016]. For example, users send 500 million tweets every day on Twitter [Stats]. These interactions may potentially have a lot of actionable insights for businesses, such as detecting sales opportunities when customers are complaining about certain products, identify users who are seeking help and route them to the support team. Manually analysis these data would have to be deemed impossible. By employing topic classification to perform sentiment analysis, one can easily make use of what people are talking about on social media, how they are doing so and track trends over time. When using the proposed relation-ontology driven topic classification for sentiment analysis, the hidden semantical meanings of the words can be considered. And it also considers the polarities of the extracted relations embedded in the texts, so that the polarities of semantical structures can be considered. The inclusion of these two factors can potentially improve the

accuracy of sentiment classification. By doing such sentiment analysis, one can easily answer questions that require a large amount of texts data, such as *"What are people complaining about when they mention a particular brand?"* or *"What are people reacting to the Brexit results?"*. These information can then be useful for product analysis, brand monitoring, customer support, and market research.

In addition, this topic classification method can be used to model customer behaviours in online deal websites and give product recommendations [Zhao et al., 2017]. The topic model can learn customers behaviours from their feedback (history of the search queries or self reports) on products as well as self-explained features (filter or conditions on the queries) [Pazzani and Billsus, 2007].

6.5 Future Work

There are many possible ways to extend this work. In terms of conceptual information used, we would like to incorporate different types of concepts besides the current ones (e.g.: *IsA* and *RelateTo*). Including more types of concepts means larger matrices and a more complicated computing process for the creation of the topic model, which can result in longer time for training. The extra concepts can potentially result in a too general topic model, and different topics are difficult to be differentiated from each other. To address this, certain rules may need to be generated to select the concepts of interest. Ontology from ConceptNet provides a hierarchical network of different types of concepts. We would like to incorporate these hierarchical relations between the ontology concepts into the topic model. These hierarchical relations can potentially form a graph-learning network to achieve a fine-tuned selection of concepts. The graph-learning network, for example, a Bayesian Network, can filter out the more relevant concepts based on the frequency and co-occurrence in the context [Andrea and Franco, 2012]. For example, Figure 6.1 shows an example of a knowledge graph using ConceptNet ontology. In this example, words such as *"PC supplier", "Engineer", "Risk Manager", etc* can be connected to form a graph-learning network by hierarchical relations between ontology concepts.

Furthermore, it is interesting to consider the inclusion of time-varying information and analyse changes in topics over time. New terms and special words are occurring every day over the Internet. The semantical meaning of these information is novel and unknown to most existing ontologies. Integrating these time-dependent knowledge with relations and concepts can result in a dynamic topic classification method, which can have great potential in the field of marketing research. Including time-varying information with the topic classification method can also be used in medical domains to analyse electronic health records (EHRs).

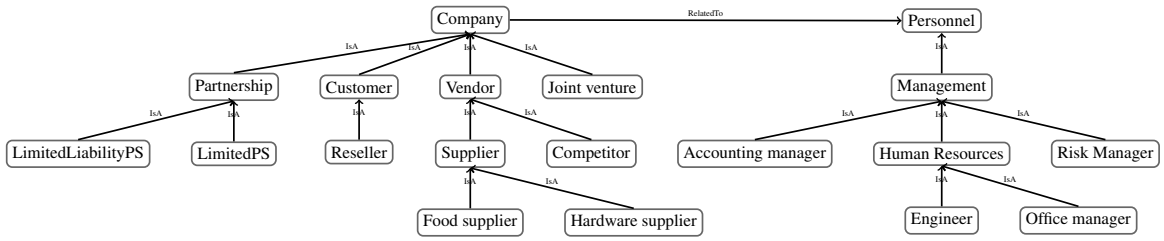


Fig. 6.1 An example of knowledge graph using ConceptNet ontology

EHRs consist of thousands of variables representing a patient admission on each day, many of which affect the patient's condition. These variables, so-called time-varying data, change over time and suggest that as a patient spends time in the hospital, his/her actual condition will vary. Furthermore, how these variables affect their conditions is likely to change over time. By including these time-varying data, the topic classification method can identify patients at high risk of acquiring a Clostridium diffcile infection (CDI).

In terms of the relation extraction algorithm, we would like to further improve the performance of the entity-based algorithm based on the discussion in Section 4.7. Three open issues are described In order to deal with more complex prepositional phrases, more general rules to understand the structure of the prepositional phrases should be designed. In order to understand the use of adjective phrases associated with the relationships, the polarity adjustment procedure can be modified. In order to identify the references to entities occurring outside a sentence, the algorithm should be able to consider the context rather than the single sentences.

In terms of applying the topic classification approach in different domains, we would like to employ different corpus for training, such as the biomedical domain and medical domain. Documents in these domains are normally unstructured in a narrative form with ambiguous terms and typographical errors, consisting of a lot of domain-specific entities and phrases. Manually annotate and classify these documents is time-consuming and requires knowledge from domain experts. Due to the lack of training dataset, this is left for future work.

References

A. B. Abacha, M. F. M. Chowdhury, A. Karanasiou, Y. Mrabet, A. Lavelli, and P. Zweigenbaum. Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification. *Journal of biomedical informatics*, 58:122–132, 2015.

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

N. Agarwal and H. Liu. Blogosphere: research issues, tools, and applications. *ACM Sigkdd Explorations Newsletter*, 10(1):18–31, 2008.

C. C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012.

S. T. Ahmed, R. Nair, C. Patel, and H. Davulcu. Bioeve: bio-molecular event extraction from text using semantic classification and dependency parsing. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 99–102. Association for Computational Linguistics, 2009.

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9(11):S2, 2008a.

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. A graph kernel for protein-protein interaction extraction. In *Proceedings of the workshop on current trends in biomedical natural language processing*, pages 1–9. Association for Computational Linguistics, 2008b.

M. Allahyari and K. Kochut. Automatic topic labeling using ontology-based topic models. pages 259–264, 12 2015. doi: 10.1109/ICMLA.2015.88.

A. A. Alurkar, S. B. Ranade, S. V. Joshi, S. S. Ranade, G. R. Shinde, P. A. Sonewar, and P. N. Mahalle. A comparative analysis and discussion of email spam classification methods using machine learning techniques. *Applied Machine Learning for Smart Data Analysis*, page 185, 2019.

B. Andrea and T. Franco. Mining bayesian networks out of ontologies. *Journal of Intelligent Information Systems*, 38(2):507–532, 2012.

S. Andreozzi, L. Magnoni, and R. Zappi. Towards the integration of storm on amazon simple storage service (s3). In *Journal of Physics: Conference Series*, volume 119, page 062011. IOP Publishing, 2008.

A. R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association, 2001.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

S. Bandari and V. V. Bulusu. Survey on ontology-based sentiment analysis of customer reviews for products and services. In *Data Engineering and Communication Technology*, pages 91–101. Springer, 2020.

M. Banko and R. C. Moore. Part of speech tagging in context. In *Proceedings of the 20th international conference on Computational Linguistics*, page 556. Association for Computational Linguistics, 2004.

D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'donovan, and R. Apweiler. The goa database in 2009—an integrated gene ontology annotation resource. *Nucleic acids research*, 37(suppl 1):D396–D403, 2009.

K. Basu, F. Shakerin, and G. Gupta. Aqua: Asp-based visual question answering. In *International Symposium on Practical Aspects of Declarative Languages*, pages 57–72. Springer, 2020.

G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114: 34–45, 2018.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

P. Bergman and S. J. Berman. *Represent yourself in court: How to prepare & try a winning case*. Nolo, 2016.

Y. Berzak, J. Kenney, C. Spadine, J. X. Wang, L. Lam, K. S. Mori, S. Garza, and B. Katz. Universal dependencies for learner english. *arXiv preprint arXiv:1605.04278*, 2016.

S. Besharati, H. Veisi, A. Darzi, and S. H. H. Saravani. A hybrid statistical and deep learning based technique for persian part of speech tagging. *Iran Journal of Computer Science*, pages 1–9, 2020.

C. M. Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.

C. Blaschke and A. Valencia. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, 17(2):14–20, 2002.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

E. E. Bolton, Y. Wang, P. A. Thiessen, and S. H. Bryant. Pubchem: integrated platform of small molecules and biological activities. *Annual reports in computational chemistry*, 4: 217–241, 2008.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

J. Breen. Jmdict: a japanese-multilingual dictionary. In *Proceedings of the Workshop on Multilingual Linguistic Ressources*, pages 71–79. Association for Computational Linguistics, 2004.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa. Workload-based software rejuvenation in cloud systems. *IEEE Transactions on Computers*, 62(6):1072–1085, 2013.

R. Bunescu and R. J. Mooney. Subsequence kernels for relation extraction. In *NIPS*, pages 171–178, 2005a.

R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.

R. Bunescu, R. Mooney, A. Ramani, and E. Marcotte. Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pages 49–56. Association for Computational Linguistics, 2006.

R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731. Association for Computational Linguistics, 2005b.

S. Burkhardt and S. Kramer. Online multi-label dependency topic models for text classification. *Machine Learning*, 107(5):859–886, 2018.

E. Cambria, R. Speer, C. Havasi, and A. Hussain. Senticnet: A publicly available semantic resource for opinion mining. In *AAAI fall symposium: commonsense knowledge*, volume 10, 2010.

B. W. Campbell, D. Mani, S. J. Curtin, R. A. Slattery, J.-M. Michno, D. R. Ort, P. J. Schaus, R. G. Palmer, J. H. Orf, and R. M. Stupar. Identical substitutions in magnesium chelatase paralogs result in chlorophyll-deficient soybean mutants. *G3: Genes, Genomes, Genetics*, 5(1):123–131, 2015a.

J. C. Campbell, A. Hindle, and E. Stroulia. Latent dirichlet allocation: extracting topics from software engineering data. In *The art and science of analyzing software data*, pages 139–159. Elsevier, 2015b.

B. Carpenter. Lingpipe for 99.99% recall of gene mentions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 307–309, 2007.

G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3): 167–174, 1992.

C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

C.-H. Chang and S.-C. Lui. Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, pages 681–688. ACM, 2001.

G. S. Chauhan and Y. K. Meena. Domsent: Domain-specific aspect term extraction in aspect-based sentiment analysis. In *Smart Systems and IoT: Innovations in Computing*, pages 103–109. Springer, 2020.

S. Chen. K-nearest neighbor algorithm optimization in text categorization. In *IOP Conference Series: Earth and Environmental Science*, volume 108, page 052074. IOP Publishing, 2018.

Y. P. Chheda, S. K. Pillai, D. G. Parikh, N. Dipayan, S. V. Shah, and G. Alaknanda. A prospective study of level iib nodal metastasis (supraretrospinal) in clinically n0 oral squamous cell carcinoma in indian population. *Indian journal of surgical oncology*, 8(2): 105–108, 2017.

Y. S. Choi. Tree pattern expression for extracting information from syntactically parsed text corpora. *Data Mining and Knowledge Discovery*, 22(1-2):211–231, 2011.

M. Chowdhury and F. Mahbub. *Improving the Effectiveness of Information Extraction from Biomedical Text*. PhD thesis, University of Trento, 2013.

Y.-C. Chu, C.-C. Hsu, C.-J. Lee, and Y.-T. Tsai. Automatic data extraction of websites using data path matching and alignment. In *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 60–64. IEEE, 2015.

K. Clark and C. D. Manning. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*, 2016a.

K. Clark and C. D. Manning. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*, 2016b.

H. Cohen and C. Lefebvre. *Handbook of categorization in cognitive science*. Elsevier, 2005.

K. B. Cohen, K. Verspoor, H. L. Johnson, C. Roeder, P. V. Ogren, W. A. Baumgartner Jr, E. White, H. Tipney, and L. Hunter. High-precision biological event extraction with a concept recognizer. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 50–58. Association for Computational Linguistics, 2009.

N. Collier, C. Nobata, and J.-i. Tsujii. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 201–207. Association for Computational Linguistics, 2000.

M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12 (ARTICLE):2493–2537, 2011.

R. L. Cooke. *The history of mathematics: A brief course*. John Wiley & Sons, 2011.

M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob. *Microsoft Azure*. Springer, 2015.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, O. E. Ajibuwa, et al. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802, 2019.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6): 391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

J. C. Denny, P. R. Irani, F. H. Wehbe, J. D. Smithers, and A. Spickard III. The knowledgemap project: development of a concept-based medical school curriculum database. In *AMIA Annual Symposium Proceedings*, volume 2003, page 195. American Medical Informatics Association, 2003.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

K. S. Dhillon, J. Singh, and J. S. Lyall. A new horizon into the pathobiology, etiology and treatment of migraine. *Medical hypotheses*, 77(1):147–151, 2011.

J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining medline: abstracts, sentences, or phrases. In *Proceedings of the pacific symposium on biocomputing*, volume 7, pages 326–337, 2002.

J. Durlach, N. Pagès, P. Bac, M. Bara, and A. Guiet-Bara. Magnesium depletion with hypo-or hyper-function of the biological clock may be involved in chronopathological forms of asthma. *Magnesium research*, 18(1):19–34, 2005.

M. Eberts and A. Ulges. Span-based joint entity and relation extraction with transformer pre-training. *arXiv preprint arXiv:1909.07755*, 2019.

N. Ehsan and A. Shakery. Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information. *Information Processing & Management*, 52(6): 1004–1017, 2016.

F. Esposito, M. Di Serafino, and P. Oresta. Atypical presentation of sacrococcygeal yolk sac tumor in infant: beware of the injuries of the gluteal region. *Journal of ultrasound*, 19(3): 227–229, 2016.

S. Farzadfar, F. Zarinkamar, and M. Hojati. Magnesium and manganese affect photosynthesis, essential oil composition and phenolic compounds of tanacetum parthenium. *Plant Physiology and Biochemistry*, 112:207–217, 2017.

T. Fayruzov, M. De Cock, C. Cornelis, and V. Hoste. Linguistic feature analysis for protein interaction extraction. *BMC bioinformatics*, 10(1):374, 2009.

R. Feldman, Y. Regev, M. Finkelstein-Landau, E. Hurvitz, and B. Kogan. Mining biomedical literature using information extraction. *Current Drug Discovery*, 2(10):19–23, 2002.

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

A. D. Fox, W. A. Baumgartner Jr, H. L. Johnson, L. E. Hunter, and D. K. Slonim. Mining protein-protein interactions from generifs with opendmap. In *Linking Literature, Information, and Knowledge for Biology*, pages 43–52. Springer, 2010.

D. A. Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

T. Frey, M. Gelhausen, and G. Saake. Categorization of concerns: a categorical program comprehension model. In *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools*, pages 73–82. ACM, 2011.

C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, 1994.

X. Fu, T. Wang, J. Li, C. Yu, and W. Liu. Improving distributed word representation and topic model by word-topic mixture model. In *Asian Conference on Machine Learning*, pages 190–205, 2016.

K. Fundel, R. Küffner, and R. Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2006.

K. Fundel, R. Küffner, and R. Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

M. Garbade. Understanding k-means clustering in machine learning. 2018.

Y. Garten, A. Coulet, and R. B. Altman. Recent progress in automatically extracting information from the pharmacogenomic literature. *Pharmacogenomics*, 11(10):1467–1489, 2010.

F. Ghamami and M. Keyvanpour. Why biomedical relation extraction is an open issue? *ICIC Express Letters, Part B: Applications*, 9:747–756, 08 2018. doi: 10.24507/icicelb.09.08.747.

S. Ghosh. *Distributed systems: an algorithmic approach*. Chapman and Hall/CRC, 2014.

D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

J. Giorgi, X. Wang, N. Sahar, W. Y. Shin, G. D. Bader, and B. Wang. End-to-end named entity recognition and relation extraction using pre-trained language models. *arXiv preprint arXiv:1912.13415*, 2019.

M. Girolami and A. Kabán. On an equivalence between plsi and lda. In *SIGIR*, volume 3, pages 433–434, 2003.

C. Giuliano, A. Lavelli, and L. Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *EACL*, volume 18, pages 401–408. Citeseer, 2006.

A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8(Oct):2265–2295, 2007.

S. Goldwater and T. Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *Annual meeting-association for computational linguistics*, volume 45, page 744. Citeseer, 2007.

M. D. Gordon and R. K. Lindsay. Toward discovery support systems: A replication, re-examination, and extension of swanson's work on literature-based discovery of a connection between raynaud's and fish oil. *Journal of the Association for Information Science and Technology*, 47(2):116–128, 1996.

S. Goryachev, M. Sordo, and Q. T. Zeng. A suite of natural language processing tools developed for the i2b2 project. In *AMIA Annual Symposium Proceedings*, volume 2006, page 931. American Medical Informatics Association, 2006.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

B. T. Grys, D. S. Lo, N. Sahin, O. Z. Kraus, Q. Morris, C. Boone, and B. J. Andrews. Machine learning and computer vision approaches for phenotypic profiling. *J Cell Biol*, 216(1):65–71, 2017.

Y. Gu, M. Gu, Y. Long, G. Xu, Z. Yang, J. Zhou, and W. Qu. An enhanced short text categorization model with deep abundant representation. *World Wide Web*, 21(6):1705–1719, 2018.

W. Guo and M. Diab. Semantic topic models: Combining word distributional statistics and dictionary definitions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 552–561. Association for Computational Linguistics, 2011.

A. Gupta, I. Banerjee, and D. L. Rubin. Automatic information extraction from unstructured mammography reports using distributed semantics. *Journal of biomedical informatics*, 2018.

J. Hakenberg. *Mining relations from the biomedical literature*. PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2010.

J. Hakenberg, R. Leaman, N. H. Vo, S. Jonnalagadda, R. Sullivan, C. Miller, L. Tari, C. Baral, and G. Gonzalez. Efficient extraction of protein-protein interactions from full-text articles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):481–494, 2010.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

E.-H. S. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *European conference on principles of data mining and knowledge discovery*, pages 424–431. Springer, 2000.

A. Handy. Amazon introduces lambda, containers at aws re: Invent. *BZ Media LLC*, 14, 2014.

Q. Hao, J. Keppens, and O. Rodrigues. A verb-based algorithm for multiple-relation extraction from single sentences. In *Proceedings of the 2017 International Conference on Information and Knowledge Engineering*, pages 115–121, 2017.

Y. Hao, X. Zhu, M. Huang, and M. Li. Discovering patterns to extract protein–protein interactions from the literature: Part ii. *Bioinformatics*, 21(15):3294–3300, 2005.

C. Havasi, R. Speer, and J. Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing*, pages 27–29. Citeseer, 2007.

M. He, Y. Wang, and W. Li. Ppi finder: a mining tool for human protein-protein interactions. *PloS one*, 4(2):e4554, 2009.

S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. Serverless computation with openlambda. In *8th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 16)*, 2016.

S. Henry and B. T. McInnes. Literature based discovery: Models, methods, and trends. *Journal of biomedical informatics*, 74:20–32, 2017.

C. Hermes Sales, D. Azevedo Nascimento, A. C. Queiroz Medeiros, K. Costa Lima, L. F. Campos Pedrosa, and C. Colli. There is chronic latent magnesium deficiency in apparently healthy university students. *Nutricion hospitalaria*, 30(1), 2014.

K. M. Hettne, R. Stierum, M. J. Schuemie, P. J. Hendriksen, B. J. Schijvenaars, E. M. v. Mulligen, J. Kleinjans, and J. A. Kors. A dictionary to identify small molecules and drugs in free text. *Bioinformatics*, 25(22):2983–2991, 2009.

R. Hida, N. Takeishi, T. Yairi, and K. Hori. Dynamic and static topic model for analyzing time-series document collections. *arXiv preprint arXiv:1805.02203*, 2018.

S. Hingmire and S. Chakraborti. Topic labeled text classification: a weakly supervised approach. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 385–394, 2014.

G. E. Hinton, T. J. Sejnowski, and T. A. Poggio. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

T. Homburg and C. Chiarcos. Word segmentation for akkadian cuneiform. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4067–4074, 2016.

D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

A. Hossein-Nezhad, A. Spira, and M. F. Holick. Influence of vitamin d status and vitamin d3 supplementation on genome wide expression of white blood cells: a randomized double-blind clinical trial. *PloS one*, 8(3):e58725, 2013.

D. Hristovski, B. Peterlin, J. A. Mitchell, and S. M. Humphrey. Using literature-based discovery to identify disease candidate genes. *International journal of medical informatics*, 74(2):289–298, 2005.

C.-I. Hsu and C. Chiu. A hybrid latent dirichlet allocation approach for topic classification. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 312–315. IEEE, 2017.

Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine learning*, 95(3):423–469, 2014.

D. Huang and H. Wu. *Mobile cloud computing: foundations and service models*. Morgan Kaufmann, 2017.

M. Huang, X. Zhu, Y. Hao, D. G. Payan, K. Qu, and M. Li. Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612, 2004.

I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 465–474, 2013.

B. L. Humphreys and D. Lindberg. The umls project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170, 1993.

P. Iannucci, M. Gupta, et al. *IBM SmartCloud: Building a cloud enabled data center*. IBM Redbooks, 2013.

Z. Iman, S. Sanner, M. R. Bouadjenek, and L. Xie. A longitudinal study of topic classification on twitter. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

Ž. Ivezić, A. J. Connolly, J. T. VanderPlas, and A. Gray. *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*. Princeton University Press, 2019.

C. Jia, X. Liang, and Y. Zhang. Cross-domain ner using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474, 2019.

R. Kabiljo, A. B. Clegg, and A. J. Shepherd. A realistic assessment of methods for extracting gene/protein interactions from free text. *BMC bioinformatics*, 10(1):233, 2009.

A. I. Kadhim. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1):273–292, 2019.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

A. Kassambara. *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. STHDA, 2017.

K. A. Kaufman. Conceptual clustering. *Encyclopedia of the Sciences of Learning*, pages 738–740, 2012.

A. S. Kavitha, P. Shivakumara, G. H. Kumar, and T. Lu. A new watershed model based system for character segmentation in degraded text lines. *AEU-International Journal of Electronics and Communications*, 71:45–52, 2017.

M. Kayed and C.-H. Chang. Fivatech: Page-level web data extraction from template pages. *IEEE transactions on knowledge and data engineering*, 22(2):249–263, 2010.

K. Khamar. Short text classification using knn based on distance function. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(4):1916–1919, 2013.

A. Khan, B. Baharudin, L. H. Lee, and K. Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1): 4–20, 2010.

A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, pages 1–62, 2020.

M. Khordad and R. E. Mercer. Identifying genotype-phenotype relationships in biomedical text. *Journal of biomedical semantics*, 8(1):57, 2017.

J. Kiefer, J. Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

B. Kim, D. Lee, J. Oh, and H. Yu. Scalable disk-based topic modeling for memory limited devices. *Information Sciences*, 516:353–369, 2020.

J. Kim, A. Mitchell, T. K. Attwood, and M. Hilario. Learning to extract relations for protein annotation. *Bioinformatics*, 23(13):i256–i263, 2007.

J. Kim, C. Bang, H. Hwang, D. Kim, C. Park, and S. Park. Ima: Identifying disease-related genes using mesh terms and association rules. *Journal of biomedical informatics*, 76: 110–123, 2017.

J.-J. Kim, Z. Zhang, J. C. Park, and S.-K. Ng. Biocontrasts: extracting and exploiting protein–protein contrastive relations from biomedical literature. *Bioinformatics*, 22(5): 597–605, 2006.

M. Kim. Detection of gene interactions based on syntactic relations. *BioMed Research International*, 2008, 2008.

S. Kim, S. Shin, I. Lee, S. Kim, R. Sriram, and B. Zhang. Pie: an online prediction system for protein–protein interactions from text. *Nucleic acids research*, 36(suppl 2):W411–W415, 2008.

S. Kim, J. Yoon, J. Yang, and S. Park. Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC bioinformatics*, 11(1):107, 2010.

E. S. Kinsky, K. Freberg, C. Kim, M. Kushin, and W. Ward. Hootsuite university: Equipping academics and future pr professionals for social media success. *Journal of Public Relations Education*, 2(1):1–18, 2016.

D. Klein and C. D. Manning. Conditional structure versus conditional estimation in nlp models. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 9–16. Association for Computational Linguistics, 2002.

E. Knorr and G. Gruman. What cloud computing really means. *InfoWorld*, 7:20–20, 2008.

Y. Ko and J. Seo. Text classification from unlabeled documents with bootstrapping and feature projection techniques. *Information Processing & Management*, 45(1):70–83, 2009.

A. Koike, Y. Niwa, and T. Takagi. Automatic extraction of gene/protein biological functions from biomedical text. *Bioinformatics*, 21(7):1227–1236, 2005.

S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

J. Kottmann, B. Margulies, G. Ingersoll, I. Drost, J. Kosin, J. Baldridge, T. Goetz, T. Morton, W. Silva, A. Autayeu, et al. Apache OpenNLP. *Online (May 2011)*, 2011.

K. Krasnashchok and S. Jouili. Improving topic quality by promoting named entities in topic modeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 247–253, 2018.

K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. F. Jones, R. Forshee, M. Walderhaug, and T. Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review. *Journal of biomedical informatics*, 73:14–29, 2017.

S. Krishnan and J. L. U. Gonzalez. Google cloud sql. In *Building Your Next Big Thing with Google Cloud Platform*, pages 159–183. Springer, 2015.

J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

G. Lample and A. Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

E. J. Lauría and A. D. March. Combining bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. *Journal of Data and Information Quality (JDIQ)*, 2(3):1–22, 2011.

A. Lavelli, F. Sebastiani, and R. Zanoli. Distributional term representations: an experimental comparison. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 615–624, 2004.

A. Lazaridou, I. Titov, and C. Sporleder. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1630–1639, 2013.

D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

Q. Le Minh, S. N. Truong, and Q. H. Bao. A pattern approach for biomedical event annotation. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 149–150. Association for Computational Linguistics, 2011.

M.-H. Lee, P. Kachroo, P. C. Pagano, J. Yanagawa, G. Wang, T. C. Walser, K. Krysan, S. Sharma, M. S. John, S. M. Dubinett, et al. Combination treatment with apricoxib and il-27 enhances inhibition of epithelial-mesenchymal transition in human lung cancer cells through a stat1 dominant pathway. *Journal of cancer science & therapy*, 6(11):468, 2014.

S.-Z. Lee, J.-i. Tsujii, and H.-C. Rim. Part-of-speech tagging based on hidden markov model assuming joint independence. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 263–269. Association for Computational Linguistics, 2000.

R. Levy and G. Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234, 2006.

C. Li, S. Chen, and Y. Qi. Filtering and classifying relevant short text with a few seed words. *Data and Information Management*, 3(3):165–186, 2019a.

J. Li, Z. Zhang, X. Li, and H. Chen. Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5):756–769, 2008.

J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Q. Li, S. Shah, X. Liu, A. Nourbakhsh, and R. Fang. Tweet topic classification using distributed language representations. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 81–88. IEEE, 2016.

X. Li, C. Li, J. Chi, J. Ouyang, and C. Li. Dataless text classification: A topic modeling approach with document manifold. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 973–982, 2018.

X. Li, Y. Meng, X. Sun, Q. Han, A. Yuan, and J. Li. Is word segmentation necessary for deep learning of chinese representations? *arXiv preprint arXiv:1905.05526*, 2019b.

X. Li, F. Yin, Z. Sun, X. Li, A. Yuan, D. Chai, M. Zhou, and J. Li. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529*, 2019c.

Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

J. Lilleberg, Y. Zhu, and Y. Zhang. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 136–140. IEEE, 2015.

J. Lin. What is cloud computing. *Class lecture Presentation*, 2008.

R. K. Lindsay and M. D. Gordon. Literature-based discovery by lexical statistics. *Journal of the Association for Information Science and Technology*, 50(7):574, 1999.

R. B. Lipton, J. M. Pavlovic, S. R. Haut, B. M. Grosberg, and D. C. Buse. Methodological issues in studying trigger factors and premonitory features of migraine. *Headache: The Journal of Head and Face Pain*, 54(10):1661–1669, 2014.

H. Liu, S. J. Bielinski, S. Sohn, S. Murphy, K. B. Wagholikar, S. R. Jonnalagadda, K. Ravikumar, S. T. Wu, I. J. Kullo, and C. G. Chute. An information extraction framework for cohort identification using electronic health records. *AMIA Summits on Translational Science Proceedings*, 2013:149, 2013.

W. Liu and L. Wang. How does dictionary size influence performance of vietnamese word segmentation? In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1079–1083, 2016.

W. Liu, P. Liu, Y. Yang, J. Yi, and Z. Zhu. A< word, part of speech> embedding model for text classification. *Expert Systems*, page e12460, 2019.

Y. Liu, Z. Liu, T.-S. Chua, and M. Sun. Topical word embeddings. In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer, 2015.

Q. Lu, J. G. Conrad, K. Al-Kofahi, and W. Keenan. Legal document clustering with built-in topic segmentation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 383–392. ACM, 2011.

M.-C. Lud and G. Widmer. Relative unsupervised discretization for association rule mining. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 148–158. Springer, 2000.

Y. Luo, Ö. Uzuner, and P. Szolovits. Bridging semantics and syntax with graph algorithms—state-of-the-art of extracting biomedical relations. *Briefings in bioinformatics*, 18(1):160–178, 2017.

M. Magrane and U. Consortium. Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011, 2011.

D. Mallampati, K. C. Shekar, and K. Ravikanth. Supervised machine learning classifier for email spam filtering. In *Innovations in Computer Science and Engineering*, pages 357–363. Springer, 2019.

C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014a.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014b.

M. Mao and M. Humphrey. A performance study on the vm startup time in the cloud. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 423–430. IEEE, 2012.

A. McCallum, D. Freitag, and F. C. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.

A. K. McCallum. Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*, 2002.

Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web*, pages 533–542, 2006.

P. Mell, T. Grance, et al. The nist definition of cloud computing. 2011.

S. Menard. *Applied logistic regression analysis*, volume 106. Sage, 2002.

R. S. Michalski, I. Bratko, M. Kubat, et al. *Machine learning and data mining: methods and applications*, volume 388. wiley New York, 1998.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, K. Chen, G. Corrado, J. Dean, L. Sutskever, and G. Zweig. word2vec. *URL https://code. google. com/p/word2vec*, 22, 2013b.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

M. Miwa and M. Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

M. Miwa and Y. Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, 2014.

M. Miwa, R. Sætre, Y. Miyao, T. Ohta, and J. Tsujii. Combining multiple layers of syntactic information for protein-protein interaction extraction. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine*, pages 101–108, 2008.

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. Protein–protein interaction extraction by leveraging multiple kernels and parsers. *International journal of medical informatics*, 78 (12):e39–e46, 2009a.

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 121–130. Association for Computational Linguistics, 2009b.

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

D. Mochihashi, T. Yamada, and N. Ueda. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 100–108. Association for Computational Linguistics, 2009.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

R. Moncayo and H. Moncayo. The womed model of benign thyroid disease: acquired magnesium deficiency due to physical and psychological stressors relates to dysfunction of oxidative phosphorylation. *BBA clinical*, 3:44–64, 2015.

T. Morton, J. Kottmann, J. Baldridge, and G. Bierner. Opennlp: A java-based nlp toolkit. EACL, 2005.

A. Moschitti. Making tree kernels practical for natural language learning. In *Eacl*, volume 113, page 24, 2006.

G. Murugesan, S. Abdulkadhar, and J. Natarajan. Distributed smoothed tree kernel for protein-protein interaction extraction from the biomedical literature. *PLoS One*, 12(11): e0187379, 2017.

A. Mykowiecka, M. Marciniak, and A. Kupść. Rule-based information extraction from patients' clinical data. *Journal of biomedical informatics*, 42(5):923–936, 2009.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

C. Nédellec. Learning language in logic-genic interaction extraction challenge. In *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*, volume 7, pages 1–7. Citeseer, 2005.

Q. L. Nguyen, D. Tikk, and U. Leser. Simple tricks for improving pattern-based information extraction from the biomedical literature. *Journal of biomedical semantics*, 1(1):9, 2010.

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang. A review of deep learning based speech synthesis. *Applied Sciences*, 9(19):4050, 2019.

Y. Niu, D. Otasek, and I. Jurisica. Evaluation of linguistic features useful in extraction of interactions from pubmed; application to annotating known, high-throughput and predicted interactions in i2d. *Bioinformatics*, 26(1):111–119, 2010.

S. M. R. Nouri, H. Li, S. Venugopal, W. Guo, M. He, and W. Tian. Autonomic decentralized elasticity based on a reinforcement learning controller for cloud applications. *Future Generation Computer Systems*, 94:765–780, 2019.

U. Ocepek, J. Rugelj, and Z. Bosnić. Improving matrix factorization recommendations for examples in cold start. *Expert Systems with Applications*, 42(19):6784–6794, 2015.

T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

D. W. Otter, J. R. Medina, and J. K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia. How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 522–531. IEEE Press, 2013.

M. Patel. When two trends fuse: Pytorch and recommender systems, 2018.

M. Paul and R. Girju. Topic modeling of research fields: An interdisciplinary perspective. In *Proceedings of the International Conference RANLP-2009*, pages 337–342, 2009.

M. Pavlinek and V. Podgorelec. Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, 80:83–93, 2017.

M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

W. Pei, T. Ge, and B. Chang. Max-margin tensor neural network for chinese word segmentation. In *ACL (1)*, pages 293–303, 2014.

N. Peinelt, D. Nguyen, and M. Liakata. tbert: Topic models and bert joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7047–7055, 2020.

D. Peleg. Distributed computing: a locality-sensitive approach, siam monographs discrete math appl. *SIAM, Philadelphia*, 2000.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

J. W. Perry, A. Kent, and M. M. Berry. Machine literature searching x. machine language; factors underlying its design and development. *Journal of the Association for Information Science and Technology*, 6(4):242–254, 1955.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

T. M. Phuong, D. Lee, and K. H. Lee. Learning rules to extract protein interactions from biomedical text. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 148–158. Springer, 2003.

C. Plake, J. Hakenberg, and U. Leser. Optimizing syntax patterns for discovering protein-protein interactions. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 195–201. ACM, 2005.

M. Y. Potrus, U. K. Ngah, and B. S. Ahmed. An evolutionary harmony search algorithm with dominant point detection for recognition-based segmentation of online arabic text recognition. *Ain Shams Engineering Journal*, 5(4):1129–1139, 2014.

D. Proux, F. Rechenmann, and L. Julliard. A pragmatic information extraction strategy for gathering data on genetic interactions. In *Ismb*, volume 8, pages 279–285, 2000.

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8 (1):50, 2007.

S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC bioinformatics*, 9(3):S6, 2008.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI*, 2018.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

K. Raja, S. Subramani, and J. Natarajan. Ppinterfinder—a mining tool for extracting causal relations on human proteins from literature. *Database*, 2013:bas052, 2013.

D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

D. Rao and B. McMahan. *Natural language processing with PyTorch: build intelligent language applications using deep learning*. " O'Reilly Media, Inc.", 2019.

A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142. Philadelphia, USA, 1996.

K. Ravi and V. Ravi. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46, 2015.

D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, and A. Jimeno. Text processing through web services: calling whatizit. *Bioinformatics*, 24(2):296–298, 2008.

D. Rebholz-Schuhmann, A. Jimeno-Yepes, M. Arregui, and H. Kirsch. Measuring prediction capacity of individual verbs for the identification of protein interactions. *Journal of biomedical informatics*, 43(2):200–207, 2010.

L. Recalde and R. Baeza-Yates. What kind of content are you prone to tweet? multi-topic preference model for tweeters. *arXiv preprint arXiv:1807.07162*, 2018.

M. Reisenbichler and T. Reutterer. Topic modeling in marketing: recent advances and research opportunities. *Journal of Business Economics*, 89(3):327–356, 2019.

Z. Ren, W. Shi, J. Wan, F. Cao, and J. Lin. Realistic and scalable benchmarking cloud file systems: Practices and lessons from alicloud. *IEEE Transactions on Parallel and Distributed Systems*, 28(11):3272–3285, 2017.

S. M. Reynolds and J. A. Bilmes. Part-of-speech tagging using virtual evidence and negative training. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 459–466. Association for Computational Linguistics, 2005.

F. Rinaldi, G. Schneider, K. Kaljurand, M. Hess, and M. Romacker. An environment for relation mining over richly annotated corpora: the case of genia. *BMC bioinformatics*, 7 (3):S3, 2006.

F. Rinaldi, G. Schneider, K. Kaljurand, M. Hess, C. Andronis, O. Konstandi, and A. Persidis. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach. *Artificial intelligence in medicine*, 39(2):127–136, 2007.

I. Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

S. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

D. Robinson. *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, SimpleDB and SQS Web Services enables you to reach business goals faster*. Emereo Pty Ltd, 2008.

A. Rosanoff, C. M. Weaver, and R. K. Rude. Suboptimal magnesium status in the united states: are the health consequences underestimated? *Nutrition reviews*, 70(3):153–164, 2012.

P. H. Rossi, M. W. Lipsey, and H. E. Freeman. *Evaluation: A systematic approach*. Sage publications, 2003.

S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

R. Sætre, K. Sagae, and J. T. Syntactic features for protein-protein interaction extraction. *LBM (Short Papers)*, 319, 2007.

R. Sagar. Openai releases gpt-3, the largest model so far". *Analytics India Magazine*, 2020.

M. Sahlgren. A brief history of word embeddings (and some clarifications). *URL: https://www. linkedin. com/pulse/brief-history-word-embeddingssome-clarifications-magnussahlgren/(visited on 11/23/2018)*, 2015.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

E. F. Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

W. S. Sarle. Neural networks and statistical models. 1994.

G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

O. Y. Saygili. *Oracle IaaS: Quick Reference Guide to Cloud Solutions*. Apress, 2017.

D. C. Schmidt, D. L. Levine, and C. Cleeland. Architectures and patterns for developing high-performance, real-time orb endsystems. In *Advances in Computers*, volume 48, pages 1–118. Elsevier, 1999.

C. Schneider. The biggest data challenges that you might not even know you have, 2016. URL https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/.

K. K. Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

N. Sebe, I. Cohen, A. Garg, and T. S. Huang. *Machine learning in computer vision*, volume 29. Springer Science & Business Media, 2005.

I. Segura-Bedmar, P. Martinez, and C. de Pablo-Sánchez. Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of biomedical informatics*, 44(5):789–804, 2011.

C. Senger, B. A. Grüning, A. Erxleben, K. Döring, H. Patel, S. Flemming, I. Merfort, and S. Günther. Mining and evaluation of molecular relationships in literature. *Bioinformatics*, 28(5):709–714, 2012.

B. Settles. Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

A. Sharma, R. Swaminathan, and H. Yang. A verb-centric approach for relationship extraction in biomedical text. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 377–385. IEEE, 2010.

D. M. Shawky and A. F. Ali. Defining a measure of cloud computing elasticity. In *2012 1st International conference on systems and computer science (ICSCS)*, pages 1–5. IEEE, 2012.

H. Shi, W. Zhan, and X. Li. A supervised fine-grained sentiment analysis system for online reviews. *Intelligent Automation & Soft Computing*, 21(4):589–605, 2015.

J. Singh and V. Gupta. A systematic review of text stemming techniques. *Artificial Intelligence Review*, 48(2):157–217, 2017.

A. Skusa, A. Rüegg, and J. Köhler. Extraction of biological interaction networks from scientific literature. *Briefings in Bioinformatics*, 6(3):263–276, 2005.

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211, 2012.

R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, 2013a.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013b.

M. Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang. Pkde4j: Entity and relation extraction for public knowledge discovery. *Journal of biomedical informatics*, 57:320–332, 2015.

R. Speer, C. Havasi, and H. Lieberman. Analogyspace: Reducing the dimensionality of common sense knowledge. In *AAAI*, volume 8, pages 548–553, 2008.

R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

P. Srinivasan. Text mining: generating hypotheses from medline. *Journal of the American Society for Information Science and Technology*, 55(5):396–413, 2004.

B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842, 2010.

W. Sriurai. Improving text categorization by using a topic model. *Advanced Computing*, 2 (6):21, 2011.

I. L. Stats. Twitter usage statistics. URL https://www.internetlivestats.com/twitter-statistics/.

W. Su, J. Wang, F. H. Lochovsky, and Y. Liu. Combining tag and value similarity for data extraction and alignment. *IEEE Transactions on knowledge and Data Engineering*, 24(7): 1186–1200, 2012.

L. V. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. S. Batra, P. V. Kamesam, and R. Kothari. Information extraction from biomedical literature: methodology, evaluation and an application. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 410–417. ACM, 2003.

K. Subramanian. Recession is good for cloud computing-microsoft agrees, 2009.

F.-T. Sun, M. Griss, O. J. Mengshoel, and Y.-T. Yeh. Latent topic analysis for predicting group purchasing behavior on the social web. 2013.

S. Swain and S. S. Sarangi. Study of various classification algorithms using data mining. *International Journal of Advanced Research in*, 2(2):110–114, 2013.

A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

A. S. Tanenbaum and M. Van Steen. *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.

S. M. Thede and M. P. Harper. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 175–182. Association for Computational Linguistics, 1999.

P. Thomas, S. Pietschmann, I. Solt, D. Tikk, and U. Leser. Not all links are equal: Exploiting dependency types for the extraction of protein-protein interactions from text. In *Proceedings of BioNLP 2011 Workshop*, pages 1–9. Association for Computational Linguistics, 2011.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

J. Tolles and W. J. Meurer. Logistic regression: relating patient characteristics to outcomes. *Jama*, 316(5):533–534, 2016.

V. I. Torvik and N. R. Smalheiser. A quantitative model for linking two disparate sets of articles in medline. *Bioinformatics*, 23(13):1658–1665, 2007.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

D. Trieschnigg, P. Pezik, V. Lee, F. De Jong, W. Kraaij, and D. Rebholz-Schuhmann. Mesh up: effective mesh text classification for improved document retrieval. *Bioinformatics*, 25 (11):1412–1418, 2009.

H. Turtle. Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1-2):5–54, 1995.

P. University. About wordnet. *WordNet*, 2010.

S. Van Landeghem, Y. Saeys, B. De Baets, and Y. Van de Peer. Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *3rd International symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 77–84. Turku Centre for Computer Sciences (TUCS), 2008.

S. Van Landeghem, T. Abeel, Y. Saeys, and Y. Van de Peer. Discriminative and informative features for biomolecular text mining with ensemble feature selection. *Bioinformatics*, 26 (18):i554–i560, 2010.

J. Varia, S. Mathew, et al. Overview of amazon web services. *Amazon Web Services*, pages 1–22, 2014.

P. Verga, E. Strubell, and A. McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv preprint arXiv:1802.10569*, 2018.

N. T. Vu, H. Adel, P. Gupta, and H. Schütze. Combining recurrent and convolutional neural networks for relation classification. *arXiv preprint arXiv:1605.07333*, 2016.

S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967a.

S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967b.

S. Wallis and G. Nelson. Knowledge discovery in grammatically analysed corpora. *Data Mining and Knowledge Discovery*, 5(4):305–335, 2001.

C. Wang, Y. Song, D. Roth, M. Zhang, and J. Han. World knowledge as indirect supervision for document clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(2):1–36, 2016.

D. Wang, M. Thint, and A. Al-Rubaie. Semi-supervised latent dirichlet allocation and its application for document classification. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 306–310. IEEE Computer Society, 2012.

H. Wang and K. Wong. Recommendation-assisted personal web. In *2013 IEEE Ninth World Congress on Services*, pages 136–140. IEEE, 2013.

H. Wang, Y. Chen, H. Kao, and S. Tsai. Inference of transcriptional regulatory network by bootstrapping patterns. *Bioinformatics*, 27(10):1422–1428, 2011.

H. Wang, M. Tan, M. Yu, S. Chang, D. Wang, K. Xu, X. Guo, and S. Potdar. Extracting multiple-relations in one-pass with pre-trained transformers. *arXiv preprint arXiv:1902.01030*, 2019.

X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, 2006.

Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, et al. Clinical information extraction applications: A literature review. *Journal of biomedical informatics*, 2017.

K. Welch and N. M. Ramadan. Mitochondria, magnesium and migraine. *Journal of the neurological sciences*, 134(1-2):9–14, 1995.

Z. I. Willis, A. S. Boyd, and M. Cecilia Di Pentima. Phototoxicity, pseudoporphyria, and photo-onycholysis due to voriconazole in a pediatric patient with leukemia and invasive aspergillosis. *Journal of the Pediatric Infectious Diseases Society*, 4(2):e22–e24, 2014.

D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl 1):D901–D906, 2008.

C. Wu and R. Buyya. *Cloud Data Centers and Cost Modeling: A complete guide to planning, designing and building a cloud data center*. Morgan Kaufmann, 2015.

Z. Wu, H. Zhu, G. Li, Z. Cui, H. Huang, J. Li, E. Chen, and G. Xu. An efficient wikipedia semantic matching approach to text document classification. *Information Sciences*, 393: 15–28, 2017.

C. Xia, C. Zhang, T. Yang, Y. Li, N. Du, X. Wu, W. Fan, F. Ma, and P. Yu. Multi-grained named entity recognition. *arXiv preprint arXiv:1906.08449*, 2019.

V. Yadav, R. Sharp, and S. Bethard. Deep affix features improve neural named entity recognizers. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 167–172, 2018.

A. Yakushiji, Y. Miyao, T. Ohta, Y. Tateisi, and J. Tsujii. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 284–292. Association for Computational Linguistics, 2006.

H. Yan, X. Qiu, and X. Huang. A graph-based model for joint chinese word segmentation and dependency parsing. *Transactions of the Association for Computational Linguistics*, 8:78–92, 2020.

J. Yang, Y. Zhang, and F. Dong. Neural reranking for named entity recognition. *arXiv preprint arXiv:1707.05127*, 2017.

J. Yang, Y. Zhang, and S. Liang. Subword encoding in lattice lstm for chinese word segmentation. *arXiv preprint arXiv:1810.12594*, 2018.

W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

K. Yono, K. Izumi, H. Sakaji, H. Matsushima, and T. Shimada. Extraction of focused topic and sentiment of financial market by using supervised topic model for price movement prediction. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–7. IEEE, 2019.

D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.

Y. Zhai and B. Liu. Structured data extraction from the web based on partial tree alignment. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1614–1628, 2006.

D. Zhang and D. Wang. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*, 2015.

J. Zhang, K. Kowsari, J. H. Harrison, J. M. Lobo, and L. E. Barnes. Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record. *IEEE Access*, 6:65333–65346, 2018a.

M. Zhang, N. Yu, and G. Fu. A simple and effective neural model for joint word segmentation and pos tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1528–1538, 2018b.

M. Zhang, Q. Wang, and G. Fu. End-to-end neural opinion extraction with a transition-based model. *Information Systems*, 80:56–63, 2019.

R. Zhang, F. Meng, Y. Zhou, and B. Liu. Relation classification via recurrent neural network with attention and tensor layers. *Big Data Mining and Analytics*, 1(3):234–244, 2018c.

T. Zhao, M. Hu, R. Rahimi, and I. King. It's about time! modeling customer behaviors as the secretary problem in daily deal websites. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3670–3679. IEEE, 2017.

W. Zheng, H. Lin, Z. Zhao, B. Xu, Y. Zhang, Z. Yang, and J. Wang. A graph kernel based on context vectors for extracting drug–drug interactions. *Journal of biomedical informatics*, 61:34–43, 2016.

D. Zhou, Y. He, and C. Kwoh. From biomedical literature to knowledge: mining protein-protein interactions. *Computational Intelligence in Biomedicine and Bioinformatics*, pages 397–421, 2008.

P. Zhou, S. Zheng, J. Xu, Z. Qi, H. Bao, and B. Xu. Joint extraction of multiple relations and entities by using a hybrid neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 135–146. Springer, 2017.

X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

P. Zweigenbaum, D. Demner-Fushman, H. Yu, and K. B. Cohen. Frontiers of biomedical text mining: current progress. *Briefings in bioinformatics*, 8(5):358–375, 2007.

Appendix A

Published Papers

1. Hao, Q., Keppens, J., and Rodrigues, O. (2017). A verb-based algorithm for multiple-relation extraction from single sentences. In Proceedings of the 2017 International Conference on Information and Knowledge Engineering, pages 115–121 Hao et al. [2017].

2. Hao, Q., Keppens, J., and Rodrigues, O. (2019). A self-training ontology-driven approach for topic classification (ST-OLDA). In Proceedings of the 2019 International Conference on Computer Science and Information Technology.